

Nana Performance Summary

P.J.Maker

December 28, 2014

This document contains some measurements for the space and time costs for the nana library. Data provided includes:

- Time cost in ns.
- Space in bytes.
- Generated assembler code.
- Results for various compiler options such as optimisation.

These test results were generated using:

- Operating System, release and cpu: Linux 3.13.0-43-generic x86_64
- CPU speed is analysed by the supplied **bogomips** program which gives a value of 945.2..995.9 BogMips . Bogomips (bogus millions of instructions per seconds) are a standard unit of measure invented by Linus Torvalds for use in Linux. They represent something vaguely related to the number of instructions executed per second.
- Note: nana should be installed using **I_DEFAULT=fast ./configure** for these measurements.

The following table contains a summary of the results:

Code	Size	Time	Options
-0			
-0			
-0			
-0			
-0			
-0			
-0			
-0			
-0			

Note:

- **assert()** is your systems standard assert macro.
- **TRAD_assert()** is the traditional implementation of assert which calls **fprintf()** and **exit()** directly.
- **I()** is the nana equivalent of **assert**.

- `DI()` is implemented using the debugger. It is very space efficient but takes longer than inline C code (such as `I()`).
- `I(A(...))` is checking that all 10 values in the array `a` are positive.
- `now()` measures the current time and returns a —double— value.
- `L()` optionally prints a debugging message.
- `DL()` is the debugger equivalent of `L()`.

Note that measurement code depends on GNU CC extensions and is not a thing of great beauty.

1 How was it measured?

See `Makefile.am` and `measure.sh` for the true story, a quick summary would be:

- The code fragments are stored one per line in a file such as `summary.tst`.
- The `measure.sh` program takes as arguments a set of compiler flags such as `-O` which are used for each line in the input file.
- The code fragment is copied 256 times by a macro inside a loop which in turn executes 1024 times. For small examples it is expected that the entire loop will fit inside the cache.
- Time is measured using the nana `now()` function.

The variables and code fragments used defined in `prelude.c` and `postlude.c`. All variables are declared `volatile` to prevent the compiler optimising access to variables.

In addition all programs are compiled with the following options:

- `-g` – debugger information is always turned on since we need it for parts of the nana library. Note that `gcc` happily optimises code with `-g` enabled.
- `-fno-defer-pop` – `gcc` by default only pops arguments off the stack after a number of calls. This option causes each call to immediately pop its arguments off the stack.

2 Detailed results

This section contains some more detailed results.

2.1 Assert

Code	Size	Time	Options
<code>assert(i >= 10);</code>	28	1ns	-O0
<code>assert(i >= 10);</code>	29	1ns	-O1
<code>assert(i >= 10);</code>	13	0ns	-O3
<code>BSD_assert(i >= 10);</code>	28	1ns	-O0
<code>BSD_assert(i >= 10);</code>	4	0ns	-O1
<code>BSD_assert(i >= 10);</code>	4	0ns	-O3
<code>TRAD_assert(i >= 10);</code>	59	1ns	-O0
<code>TRAD_assert(i >= 10);</code>	63	1ns	-O1
<code>TRAD_assert(i >= 10);</code>	13	0ns	-O3
<code>I(i >= 10);</code>	28	1ns	-O0
<code>I(i >= 10);</code>	29	1ns	-O1
<code>I(i >= 10);</code>	13	0ns	-O3
<code>DI(i >= 10);</code>	10	13.5us	-O0
<code>DI(i >= 10);</code>	10	13.3us	-O1
<code>DI(i >= 10);</code>	10	13.4us	-O3

2.2 Quantifiers

Code	Size	Time	Options
<code>I(A(char *p = str, *p != '\0', p++, islower(*p)));</code>	130	46ns	-O0
<code>I(A(char *p = str, *p != '\0', p++, islower(*p)));</code>	50	13ns	-O1
<code>I(A(char *p = str, *p != '\0', p++, islower(*p)));</code>	69	9ns	-O3
<code>I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));</code>	174	418ns	-O0
<code>I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));</code>	84	125ns	-O1
<code>I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));</code>	220	76ns	-O3

2.3 Log

Code	Size	Time	Options
printf("helloworldn");	15	-O0	
printf("helloworldn");	20	-O1	
printf("helloworldn");	-4529	-O3	
L("helloworldn");	30	-O0	
L("helloworldn");	27	-O1	
L("helloworldn");	-7089	-O3	
DL("helloworldn");	10	2.9us	-O0
DL("helloworldn");	10	2.9us	-O1
DL("helloworldn");	10	2.9us	-O3
gi = 0; LG(gi & 0x10, "helloworldn");	53	1ns	-O0
gi = 0; LG(gi & 0x10, "helloworldn");	47	1ns	-O1
gi = 0; LG(gi & 0x10, "helloworldn");	24	0ns	-O3
gi = ~0; LG(gi & 0x10, "helloworldn");	53	-O0	
gi = ~0; LG(gi & 0x10, "helloworldn");	47	-O1	
gi = ~0; LG(gi & 0x10, "helloworldn");	24	-O3	
LHP(fprintf,log,"helloworldn");	27	26ns	-O0
LHP(fprintf,log,"helloworldn");	23	27ns	-O1
LHP(fprintf,log,"helloworldn");	-6065	26ns	-O3
LHP(L_buffer_printf,buf,"helloworldn");	22	78ns	-O0
LHP(L_buffer_printf,buf,"helloworldn");	18	74ns	-O1
LHP(L_buffer_printf,buf,"helloworldn");	-4014	72ns	-O3
LHP(syslog,LOG_USER,"helloworldn");	20	3.5us	-O0
LHP(syslog,LOG_USER,"helloworldn");	25	3.9us	-O1
LHP(syslog,LOG_USER,"helloworldn");	-5809	3.6us	-O3

2.4 Nop

Code	Size	Time	Options
asm("");	0	0ns	-O0
asm("");	0	0ns	-O1
asm("");	0	0ns	-O3
asm("nop");	1	0ns	-O0
asm("nop");	1	0ns	-O1
asm("nop");	1	0ns	-O3
asm("nop;nop;");	2	0ns	-O0
asm("nop;nop;");	2	0ns	-O1
asm("nop;nop;");	2	0ns	-O3
asm("nop;nop;nop;");	3	0ns	-O0
asm("nop;nop;nop;");	3	0ns	-O1
asm("nop;nop;nop;");	3	0ns	-O3
asm("nop;nop;nop;nop;");	4	0ns	-O0
asm("nop;nop;nop;nop;");	4	0ns	-O1
asm("nop;nop;nop;nop;");	4	0ns	-O3
asm("nop;nop;nop;nop;nop;");	5	1ns	-O0
asm("nop;nop;nop;nop;nop;");	5	0ns	-O1
asm("nop;nop;nop;nop;nop;");	5	0ns	-O3

2.5 C Operations

Code	Size	Time	Options
i = 4;	7	0ns	-O0
i = 4;	8	0ns	-O1
i = 4;	8	0ns	-O3
gi = 11;	10	0ns	-O0
gi = 11;	10	0ns	-O1
gi = 11;	10	0ns	-O3
f = 12.0;	9	0ns	-O0
f = 12.0;	14	0ns	-O1
f = 12.0;	8	0ns	-O3
gf = 12.0;	12	0ns	-O0
gf = 12.0;	16	0ns	-O1
gf = 12.0;	10	0ns	-O3
i++;	9	2ns	-O0
i++;	11	2ns	-O1
i++;	11	2ns	-O3
gi++;	15	2ns	-O0
gi++;	15	2ns	-O1
gi++;	15	2ns	-O3
j = a[i];	15	0ns	-O0
j = a[i];	17	1ns	-O1
j = a[i];	16	0ns	-O3

2.6 Data cache testing

These are just some tests using a large array which should hopefully exceed the size of the D-cache on your machine.

Code	Size	Time	Options
-DNT=16			
-DNT=16			
-DNT=16			
-DNT=16			
-DNT=16			
-DNT=16			
-DNT=16			
-DNT=16			

3 Code

This section contains a listing of all the generated code fragments.

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
movl    $.LC4, %edi
movl    $0, %eax
call    printf
        .loc 1 120 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:

```
.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    movl    $0, %eax
    call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
    .loc 1 120 0
```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```
.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    xorl    %eax, %eax
    call    __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 120 0
```

- `L("helloworldn");` with `gcc -g -O0` produces:

```
    .loc 1 119 0
#NO_APP
    movq    stdout(%rip), %rax
    movq    %rax, %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
    .loc 1 120 0
```

- `L("helloworldn");` with `gcc -g -O1` produces:

```
.LVL5:
#NO_APP
.LBB526:
```

```

.LBB527:
    .loc 2 97 0
    movq    stdout(%rip), %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL6:
.LBE527:
.LBE526:
    .loc 1 120 0

```

- L("helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 97 0
    movq    stdout(%rip), %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 120 0

```

- DL("helloworldn"); with gcc -g -O0 produces:

```

    .loc 1 119 0
#NO_APP
    movl    $0, _dl_target(%rip)
    .loc 1 120 0

```

- DL("helloworldn"); with gcc -g -O1 produces:

```

    .loc 1 119 0
#NO_APP
    movl    $0, _dl_target(%rip)
    .loc 1 120 0

```

- DL("helloworldn"); with gcc -g -O3 produces:

```

    .loc 1 119 0
#NO_APP
    movl    $0, _dl_target(%rip)
    .loc 1 120 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 120 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al

```



```

        jne      .L1045
.L5:
        .loc 1 120 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl     $-1, gi(%rip)
        movl     gi(%rip), %eax
        andl     $16, %eax
        testl    %eax, %eax
        je       .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movq     stdout(%rip), %rax
        movq     %rax, %rcx
        movl     $11, %edx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
.L5:
        .loc 1 120 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl     $-1, gi(%rip)
        movl     gi(%rip), %eax
        testb    $16, %al
        je       .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq     stdout(%rip), %rcx
        movl     $11, %edx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 119 0

```

```

#NO_APP
    movl    $-1, gi(%rip)
    movl    gi(%rip), %eax
    testb   $16, %al
    jne     .L1045
.L5:
    .loc 1 120 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O0 produces:

```

    .loc 1 119 0
#NO_APP
    movq    -32(%rbp), %rax
    movq    %rax, %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
    .loc 1 120 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 97 0
    movq    %rbx, %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL6:
.LBE527:
.LBE526:
    .loc 1 120 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 97 0
    movl    $11, %edx
    movq    %rbx, %rcx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite

```

```
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 120 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```
    .loc 1 119 0
#NO_APP
    movq    -40(%rbp), %rax
    movl    $.LC4, %esi
    movq    %rax, %rdi
    movl    $0, %eax
    call    L_buffer_printf
    .loc 1 120 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```
    .loc 1 119 0
#NO_APP
    movl    $.LC4, %esi
    movq    %rbx, %rdi
    movl    $0, %eax
    call    L_buffer_printf
.LVL6:
    .loc 1 120 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```
    .loc 1 119 0
#NO_APP
    movl    $.LC4, %esi
    movq    %rbx, %rdi
    xorl    %eax, %eax
    call    L_buffer_printf
.LVL279:
    .loc 1 120 0
```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O0 produces:

```
    .loc 1 119 0
#NO_APP
    movl    $.LC4, %esi
    movl    $8, %edi
    movl    $0, %eax
    call    syslog
    .loc 1 120 0
```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
    .loc 3 31 0
    movl    $.LC4, %edx
    movl    $1, %esi
    movl    $8, %edi
    movl    $0, %eax
    call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
    .loc 1 120 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 3 31 0
    movl    $.LC4, %edx
    movl    $1, %esi
    movl    $8, %edi
    xorl    %eax, %eax
    call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 120 0

```

- printf("helloworldn"); with gcc -g -O0 produces:

```

    .loc 1 119 0
#NO_APP
    movl    $.LC4, %edi
    movl    $0, %eax
    call    printf
    .loc 1 120 0

```

- printf("helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 104 0

```

```

        movl    $.LC4, %esi
        movl    $1, %edi
        movl    $0, %eax
        call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 104 0
        movl    $.LC4, %esi
        movl    $1, %edi
        xorl    %eax, %eax
        call    __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 120 0

```

- `L("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 120 0

```

- `L("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite

```

```
.LVL6:
.LBE527:
.LBE526:
    .loc 1 120 0
```

- L("helloworldn"); with gcc -g -O3 produces:

```
.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 97 0
    movq    stdout(%rip), %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 120 0
```

- DL("helloworldn"); with gcc -g -O0 produces:

```
    .loc 1 119 0
#NO_APP
    movl    $0, _dl_target(%rip)
    .loc 1 120 0
```

- DL("helloworldn"); with gcc -g -O1 produces:

```
    .loc 1 119 0
#NO_APP
    movl    $0, _dl_target(%rip)
    .loc 1 120 0
```

- DL("helloworldn"); with gcc -g -O3 produces:

```
    .loc 1 119 0
#NO_APP
    movl    $0, _dl_target(%rip)
    .loc 1 120 0
```

- gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O0 produces:

```
    .loc 1 119 0
#NO_APP
    movl    $0, gi(%rip)
    movl    gi(%rip), %eax
```

```

        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 120 0 is_stmt 1

```

- `assert(i >= 2);` with `gcc -g -O` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $1, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 120 0 is_stmt 1

```

- `TRAD_assert(i >= 2);` with `gcc -g -O` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $1, %eax
        jg      .L5
.LVL5:
.LBB783:
.LBB784:
.LBB785:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $119, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk

```

```

.LVL6:
.LBE785:
.LBE784:
    .loc 1 119 0 discriminator 1
    movl    $1, %edi
    call    exit
.LVL7:
.L5:
.LBE783:
    .loc 1 120 0

```

- `I(i >= 2);` with `gcc -g -O` produces:

```

    .loc 1 119 0
#NO_APP
    movl    24(%rsp), %eax
    cmpl    $1, %eax
    jg      .L5
    .loc 1 119 0 is_stmt 0 discriminator 1
    movl    $119, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.LVL5:
.L5:
    .loc 1 120 0 is_stmt 1

```

- `DI(i >= 2);` with `gcc -g -O` produces:

```

    .loc 1 119 0
#NO_APP
    movl    $0, _di_target(%rip)
    .loc 1 120 0

```

- `I(A(int i=0, i!=10, i++, a[i]>=0));` with `gcc -g -O` produces:

```

.LVL5:
#NO_APP
.LBB269:
    .loc 1 119 0
    cmpl    $0, a(%rip)
    js      .L5
    movl    $a+4, %eax
    movl    $a+40, %edx
.LVL6:
.L7:
    .loc 1 119 0 is_stmt 0 discriminator 2
    cmpl    $0, (%rax)
    js      .L5

```



```

        addq    $4, %rax
        cmpq    %rdx, %rax
        jne     .L7
.LVL7:
.L6:
.LBE269:
        .loc 1 120 0 is_stmt 1

```

- `d = now();` with `gcc -g -O` produces:

```

        .loc 1 119 0
#NO_APP
        call    now
.LVL5:
        movsd   %xmm0, 24(%rsp)
        .loc 1 120 0

```

- `printf("helloworldn");` with `gcc -g -O` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 104 0
        movl    $.LC4, %esi
        movl    $1, %edi
        movl    $0, %eax
        call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `L("helloworldn");` with `gcc -g -O` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `DL("helloworldn");` with `gcc -g -O` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 120 0

```

- `I(A(int i=0, i < 1*1024, i++, za[i] == 0));` with `gcc -g -DNT=16` produces:

```

#NO_APP
.LBB2:
        .loc 1 119 0
        movl    $1, -2104(%rbp)
        movl    $0, -2100(%rbp)
        jmp     .L5
.L8:
        .loc 1 119 0 is_stmt 0 discriminator 2
        movl    -2100(%rbp), %eax
        cltq
        movl    za(,%rax,4), %eax
        testl   %eax, %eax
        je      .L6
        .loc 1 119 0 discriminator 1
        movl    $0, -2104(%rbp)
        jmp     .L7
.L6:
        .loc 1 119 0 discriminator 2
        addl    $1, -2100(%rbp)
.L5:
        .loc 1 119 0 discriminator 1
        cmpl    $1023, -2100(%rbp)
        jle     .L8
.L7:
        movl    -2104(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L9
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L9:
        .loc 1 120 0 is_stmt 1

```

- `I(A(int i=0, i < 2*1024, i++, za[i] == 0));` with `gcc -g -DNT=16` produces:

```

#NO_APP

```

```

.LBB2:
    .loc 1 119 0
    movl    $1, -2104(%rbp)
    movl    $0, -2100(%rbp)
    jmp     .L5
.L8:
    .loc 1 119 0 is_stmt 0 discriminator 2
    movl    -2100(%rbp), %eax
    cltq
    movl    za(,%rax,4), %eax
    testl   %eax, %eax
    je      .L6
    .loc 1 119 0 discriminator 1
    movl    $0, -2104(%rbp)
    jmp     .L7
.L6:
    .loc 1 119 0 discriminator 2
    addl    $1, -2100(%rbp)
.L5:
    .loc 1 119 0 discriminator 1
    cmpl    $2047, -2100(%rbp)
    jle     .L8
.L7:
    movl    -2104(%rbp), %eax
.LBE2:
    testl   %eax, %eax
    jne     .L9
    movl    $119, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.L9:
    .loc 1 120 0 is_stmt 1

```

- `I(A(int i=0, i < 4*1024, i++, za[i] == 0));` with `gcc -g -DNT=16` produces:

```
#NO_APP
```

```

.LBB2:
    .loc 1 119 0
    movl    $1, -2104(%rbp)
    movl    $0, -2100(%rbp)
    jmp     .L5
.L8:
    .loc 1 119 0 is_stmt 0 discriminator 2
    movl    -2100(%rbp), %eax
    cltq
    movl    za(,%rax,4), %eax
    testl   %eax, %eax
    je      .L6

```

```

        .loc 1 119 0 discriminator 1
        movl    $0, -2104(%rbp)
        jmp     .L7
.L6:
        .loc 1 119 0 discriminator 2
        addl    $1, -2100(%rbp)
.L5:
        .loc 1 119 0 discriminator 1
        cmpl    $4095, -2100(%rbp)
        jle     .L8
.L7:
        movl    -2104(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L9
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L9:
        .loc 1 120 0 is_stmt 1

```

- I(A(int i=0, i < 8*1024, i++, za[i] == 0)); with gcc -g -DNT=16 produces:

```

#NO_APP
.LBB2:
        .loc 1 119 0
        movl    $1, -2104(%rbp)
        movl    $0, -2100(%rbp)
        jmp     .L5
.L8:
        .loc 1 119 0 is_stmt 0 discriminator 2
        movl    -2100(%rbp), %eax
        cltq
        movl    za(,%rax,4), %eax
        testl   %eax, %eax
        je      .L6
        .loc 1 119 0 discriminator 1
        movl    $0, -2104(%rbp)
        jmp     .L7
.L6:
        .loc 1 119 0 discriminator 2
        addl    $1, -2100(%rbp)
.L5:
        .loc 1 119 0 discriminator 1
        cmpl    $8191, -2100(%rbp)
        jle     .L8
.L7:
        movl    -2104(%rbp), %eax

```

```

.LBE2:
    testl    %eax, %eax
    jne      .L9
    movl     $119, %edx
    movl     $.LC4, %esi
    movl     $.LC5, %edi
    call     _I_default_handler
.L9:
    .loc 1 120 0 is_stmt 1

```

- I(A(int i=0, i < 16*1024, i++, za[i] == 0)); with gcc -g -DNT=16 produces:

```

#NO_APP
.LBB2:
    .loc 1 119 0
    movl     $1, -2104(%rbp)
    movl     $0, -2100(%rbp)
    jmp      .L5
.L8:
    .loc 1 119 0 is_stmt 0 discriminator 2
    movl     -2100(%rbp), %eax
    cltq
    movl     za(,%rax,4), %eax
    testl    %eax, %eax
    je       .L6
    .loc 1 119 0 discriminator 1
    movl     $0, -2104(%rbp)
    jmp      .L7
.L6:
    .loc 1 119 0 discriminator 2
    addl     $1, -2100(%rbp)
.L5:
    .loc 1 119 0 discriminator 1
    cmpl     $16383, -2100(%rbp)
    jle      .L8
.L7:
    movl     -2104(%rbp), %eax
.LBE2:
    testl    %eax, %eax
    jne      .L9
    movl     $119, %edx
    movl     $.LC4, %esi
    movl     $.LC5, %edi
    call     _I_default_handler
.L9:
    .loc 1 120 0 is_stmt 1

```

- I(A(int i=0, i < 32*1024, i++, za[i] == 0)); with gcc -g -DNT=16 produces:

```

#NO_APP
.LBB2:
    .loc 1 119 0
    movl    $1, -2104(%rbp)
    movl    $0, -2100(%rbp)
    jmp     .L5
.L8:
    .loc 1 119 0 is_stmt 0 discriminator 2
    movl    -2100(%rbp), %eax
    cltq
    movl    za(,%rax,4), %eax
    testl   %eax, %eax
    je      .L6
    .loc 1 119 0 discriminator 1
    movl    $0, -2104(%rbp)
    jmp     .L7
.L6:
    .loc 1 119 0 discriminator 2
    addl    $1, -2100(%rbp)
.L5:
    .loc 1 119 0 discriminator 1
    cmpl    $32767, -2100(%rbp)
    jle     .L8
.L7:
    movl    -2104(%rbp), %eax
.LBE2:
    testl   %eax, %eax
    jne     .L9
    movl    $119, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.L9:
    .loc 1 120 0 is_stmt 1

```

- I(A(int i=0, i < 64*1024, i++, za[i] == 0)); with gcc -g -DNT=16 produces:

```

#NO_APP
.LBB2:
    .loc 1 119 0
    movl    $1, -2104(%rbp)
    movl    $0, -2100(%rbp)
    jmp     .L5
.L8:
    .loc 1 119 0 is_stmt 0 discriminator 2
    movl    -2100(%rbp), %eax
    cltq
    movl    za(,%rax,4), %eax
    testl   %eax, %eax

```

```

        je      .L6
        .loc 1 119 0 discriminator 1
        movl    $0, -2104(%rbp)
        jmp     .L7
.L6:
        .loc 1 119 0 discriminator 2
        addl    $1, -2100(%rbp)
.L5:
        .loc 1 119 0 discriminator 1
        cmpl    $65535, -2100(%rbp)
        jle     .L8
.L7:
        movl    -2104(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L9
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L9:
        .loc 1 120 0 is_stmt 1

```

- I(A(int i=0, i < 128*1024, i++, za[i] == 0)); with gcc -g -DNT=16 produces:

```

#NO_APP
.LBB2:
        .loc 1 119 0
        movl    $1, -2104(%rbp)
        movl    $0, -2100(%rbp)
        jmp     .L5
.L8:
        .loc 1 119 0 is_stmt 0 discriminator 2
        movl    -2100(%rbp), %eax
        cltq
        movl    za(,%rax,4), %eax
        testl   %eax, %eax
        je      .L6
        .loc 1 119 0 discriminator 1
        movl    $0, -2104(%rbp)
        jmp     .L7
.L6:
        .loc 1 119 0 discriminator 2
        addl    $1, -2100(%rbp)
.L5:
        .loc 1 119 0 discriminator 1
        cmpl    $131071, -2100(%rbp)
        jle     .L8
.L7:

```

```

        movl    -2104(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L9
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L9:
        .loc 1 120 0 is_stmt 1

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O3` produces:

```

        .loc 1 119 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```


- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("");` with `gcc -g -O0` produces:

```
.loc 1 119 0
```

- `asm("");` with `gcc -g -O1` produces:

```
.loc 1 119 0
```

- `asm("");` with `gcc -g -O3` produces:

```
.loc 1 119 0
```

- `asm("nop");` with `gcc -g -O0` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O1` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O3` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O3` produces:

```

        .loc 1 119 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 120 0

```

- i = 4; with gcc -g -O1 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 120 0

```

- i = 4; with gcc -g -O3 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 120 0

```

- gi = 11; with gcc -g -O0 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 120 0

```

- gi = 11; with gcc -g -O1 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 120 0

```

- gi = 11; with gcc -g -O3 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 120 0

```

- f = 12.0; with gcc -g -O0 produces:

```

        .loc 1 119 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 120 0

```

- f = 12.0; with gcc -g -O1 produces:

```

        .loc 1 119 0
#NO_APP
        movss    .LC4(%rip), %xmm0
        movss    %xmm0, 28(%rsp)
        .loc 1 120 0

```

- `f = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl     $0x41400000, 28(%rsp)
        .loc 1 120 0

```

- `gf = 12.0;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl     .LC4(%rip), %eax
        movl     %eax, gf(%rip)
        .loc 1 120 0

```

- `gf = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movss    .LC4(%rip), %xmm0
        movss    %xmm0, gf(%rip)
        .loc 1 120 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl     $0x41400000, gf(%rip)
        .loc 1 120 0

```

- `i++;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl     -56(%rbp), %eax
        addl     $1, %eax
        movl     %eax, -56(%rbp)
        .loc 1 120 0

```

- `i++;` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 120 0

```

- `i++`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 120 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 120 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 120 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 120 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 120 0

```


- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 120 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movslq  20(%rsp), %rax
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 120 0

```

- `assert(i >= 10)`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 120 0 is_stmt 1

```

- `assert(i >= 10)`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 120 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 120 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $119, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 120 0 is_stmt 1

```

- `BSD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 120 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 120 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
.LBB2:
        .loc 1 119 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax

```

```

        movl    $.LC4, %r8d
        movl    $119, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
.LBE2:
        .loc 1 120 0 is_stmt 1

```

- `TRAD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB783:
.LBB784:
.LBB785:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $119, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE785:
.LBE784:
        .loc 1 119 0 discriminator 1
        movl    $1, %edi
        call    exit
.LVL7:
.L5:
.LBE783:
        .loc 1 120 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax

```

```

    cmpl    $9, %eax
    jle     .L21
    .loc 1 120 0

```

- I(i >= 10); with gcc -g -O0 produces:

```

    .loc 1 119 0
#NO_APP
    movl    -56(%rbp), %eax
    cmpl    $9, %eax
    jg      .L5
    .loc 1 119 0 is_stmt 0 discriminator 1
    movl    $119, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.L5:
    .loc 1 120 0 is_stmt 1

```

- I(i >= 10); with gcc -g -O1 produces:

```

    .loc 1 119 0
#NO_APP
    movl    24(%rsp), %eax
    cmpl    $9, %eax
    jg      .L5
    .loc 1 119 0 is_stmt 0 discriminator 1
    movl    $119, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.LVL5:
.L5:
    .loc 1 120 0 is_stmt 1

```

- I(i >= 10); with gcc -g -O3 produces:

```

    .loc 1 119 0
#NO_APP
    movl    24(%rsp), %eax
    cmpl    $9, %eax
    jle     .L274
.L5:
    .loc 1 120 0

```

- DI(i >= 10); with gcc -g -O0 produces:

```

    .loc 1 119 0
#NO_APP
    movl    $0, _di_target(%rip)
    .loc 1 120 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 120 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 120 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O0` produces:

```

#NO_APP
.LBB2:
        .loc 1 119 0
        movl    $1, -3132(%rbp)
        movq    $str, -2096(%rbp)
        jmp     .L5

.L8:
        .loc 1 119 0 is_stmt 0 discriminator 2
        call    __ctype_b_loc
        movq    (%rax), %rdx
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        movsbq  %al, %rax
        addq    %rax, %rax
        addq    %rdx, %rax
        movzwl  (%rax), %eax
        movzwl  %ax, %eax
        andl    $512, %eax
        testl   %eax, %eax
        jne     .L6
        .loc 1 119 0 discriminator 1
        movl    $0, -3132(%rbp)
        jmp     .L7

.L6:
        .loc 1 119 0 discriminator 2
        addq    $1, -2096(%rbp)

.L5:
        .loc 1 119 0 discriminator 1
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        testb   %al, %al
        jne     .L8

```

```

.L7:
    movl    -3132(%rbp), %eax
.LBE2:
    testl   %eax, %eax
    jne     .L9
    movl    $119, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.L9:
    .loc 1 120 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB269:
    .loc 1 119 0
    movzbl  str(%rip), %ebx
    testb   %bl, %bl
    je      .L5
    call    __ctype_b_loc
.LVL6:
    movq    (%rax), %rdx
    movl    $str, %eax
.LVL7:
.L7:
    .loc 1 119 0 is_stmt 0 discriminator 2
    movsbq   %bl, %rbx
    testb    $2, 1(%rdx,%rbx,2)
    je      .L6
    addq     $1, %rax
.LVL8:
    movzbl   (%rax), %ebx
    testb    %bl, %bl
    jne     .L7
.LVL9:
.L5:
.LBE269:
    .loc 1 120 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O3` produces:

```

.LVL5:
#NO_APP
.LBB271:
    .loc 1 119 0

```

```

        movsbq  str(%rip), %rbx
        testb   %b1, %b1
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
        jmp     .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 119 0 is_stmt 0 discriminator 2
        addq    $1, %rax
.LVL8:
        movsbq  (%rax), %rbx
        testb   %b1, %b1
        je      .L5
.LVL9:
.L7:
        testb   $2, 1(%rdx,%rbx,2)
        jne     .L2074
.LVL10:
.LBE271:
        .loc 1 119 0
        movl    $119, %edx
        movl    $.LC11, %esi
        movl    $.LC12, %edi
        call    _I_default_handler
.LVL11:
.L5:
        .loc 1 120 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
        .loc 1 119 0
        movl    $1, -5188(%rbp)
        movl    $0, -5184(%rbp)
        jmp     .L5
.L11:
.LBB3:
        .loc 1 119 0 is_stmt 0 discriminator 2
        movq    $0, -2096(%rbp)
        movl    $0, -5180(%rbp)
        jmp     .L6
.L8:
        movl    -5184(%rbp), %eax

```

```

        cltq
        movl    a(,%rax,4), %edx
        movl    -5180(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        cmpl    %eax, %edx
        jne     .L7
        .loc 1 119 0 discriminator 1
        addq    $1, -2096(%rbp)
.L7:
        .loc 1 119 0 discriminator 2
        addl    $1, -5180(%rbp)
.L6:
        .loc 1 119 0 discriminator 1
        cmpl    $9, -5180(%rbp)
        jle     .L8
        .loc 1 119 0 discriminator 3
        movq    -2096(%rbp), %rax
.LBE3:
        cmpq    $1, %rax
        je      .L9
        .loc 1 119 0 discriminator 1
        movl    $0, -5188(%rbp)
        jmp     .L10
.L9:
        .loc 1 119 0 discriminator 2
        addl    $1, -5184(%rbp)
.L5:
        .loc 1 119 0 discriminator 1
        cmpl    $9, -5184(%rbp)
        jle     .L11
.L10:
        movl    -5188(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L12
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L12:
        .loc 1 120 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 119 0

```



```

        movl    $0, %ecx
.LBE526:
        .loc 1 118 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 119 0 discriminator 2
        movslq  %ecx, %rdi
        movslq  %eax, %rsi
        movl    a(,%rsi,4), %ebx
        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:
        addl    $1, %eax
.LVL10:
        cmpl    $9, %eax
        jle     .L5
.LBE527:
        .loc 1 119 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 119 0 discriminator 2
        addl    $1, %ecx
.LVL11:
        cmpl    $9, %ecx
        jle     .L1295
.LVL12:
.L1038:
.LBE528:
        .loc 1 120 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB528:
.LBB529:
        .loc 1 119 0

```

```

        movl    a(%rip), %r13d
        movl    a+4(%rip), %r12d
        movl    $a, %ecx
        movl    a+8(%rip), %ebp
        movl    a+12(%rip), %ebx
        movl    a+16(%rip), %r11d
        movl    a+20(%rip), %r10d
        movl    a+24(%rip), %r9d
        movl    a+28(%rip), %r8d
        movl    a+32(%rip), %edi
        movl    a+36(%rip), %esi
.LVL6:
.L5:
        movl    (%rcx), %edx
        xorl    %eax, %eax
        cmpl    %r13d, %edx
        sete    %al
.LVL7:
        xorl    %r14d, %r14d
        cmpl    %r12d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL8:
        xorl    %r14d, %r14d
        cmpl    %ebp, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL9:
        xorl    %r14d, %r14d
        cmpl    %ebx, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL10:
        xorl    %r14d, %r14d
        cmpl    %r11d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL11:
        xorl    %r14d, %r14d
        cmpl    %r10d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL12:
        xorl    %r14d, %r14d
        cmpl    %r9d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL13:
        xorl    %r14d, %r14d
        cmpl    %r8d, %edx

```

```

        sete    %r14b
        addq    %r14, %rax
.LVL14:
        xorl    %r14d, %r14d
        cmpl    %edi, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL15:
        cmpl    %esi, %edx
        sete    %dl
        movzbl  %dl, %edx
        addq    %rdx, %rax
.LVL16:
.LBE529:
        cmpq    $1, %rax
        jne     .L4164
        addq    $4, %rcx
        .loc 1 119 0 is_stmt 0 discriminator 2
        cmpq    $a+40, %rcx
        jne     .L5
.LVL17:
.L6:
.LBE528:
        .loc 1 120 0 is_stmt 1

```

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $.LC4, %edi
        movl    $0, %eax
        call    printf
        .loc 1 120 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 104 0
        movl    $.LC4, %esi
        movl    $1, %edi
        movl    $0, %eax
        call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```
.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    xorl    %eax, %eax
    call    __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 120 0
```

- `L("helloworldn");` with `gcc -g -O0` produces:

```
    .loc 1 119 0
#NO_APP
    movq    stdout(%rip), %rax
    movq    %rax, %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
    .loc 1 120 0
```

- `L("helloworldn");` with `gcc -g -O1` produces:

```
.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 97 0
    movq    stdout(%rip), %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL6:
.LBE527:
.LBE526:
    .loc 1 120 0
```

- `L("helloworldn");` with `gcc -g -O3` produces:

```
.LVL277:
#NO_APP
```

```

.LBB1050:
.LBB1051:
    .loc 2 97 0
    movq    stdout(%rip), %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 120 0

```

- DL("helloworldn"); with gcc -g -O0 produces:

```

    .loc 1 119 0
#NO_APP
    movl    $0, _dl_target(%rip)
    .loc 1 120 0

```

- DL("helloworldn"); with gcc -g -O1 produces:

```

    .loc 1 119 0
#NO_APP
    movl    $0, _dl_target(%rip)
    .loc 1 120 0

```

- DL("helloworldn"); with gcc -g -O3 produces:

```

    .loc 1 119 0
#NO_APP
    movl    $0, _dl_target(%rip)
    .loc 1 120 0

```

- gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O0 produces:

```

    .loc 1 119 0
#NO_APP
    movl    $0, gi(%rip)
    movl    gi(%rip), %eax
    andl    $16, %eax
    testl   %eax, %eax
    je      .L5
    .loc 1 119 0 is_stmt 0 discriminator 1
    movq    stdout(%rip), %rax
    movq    %rax, %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi

```

```

        call    fwrite
.L5:
        .loc 1 120 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 120 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax

```

```

        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 120 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 120 0

```

- `LHP(fprintf,log,"helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movq    -32(%rbp), %rax
        movq    %rax, %rcx
        movl    $11, %edx

```

```

        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 120 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    %rbx, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movl    $11, %edx
        movq    %rbx, %rcx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 120 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 119 0
#NO_APP
        movq    -40(%rbp), %rax
        movl    $.LC4, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    L_buffer_printf
        .loc 1 120 0

```


- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        movl    $0, %eax
        call    L_buffer_printf
.LVL6:
        .loc 1 120 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        xorl    %eax, %eax
        call    L_buffer_printf
.LVL279:
        .loc 1 120 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $.LC4, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    syslog
        .loc 1 120 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O3` produces:

```
.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 3 31 0
    movl    $.LC4, %edx
    movl    $1, %esi
    movl    $8, %edi
    xorl    %eax, %eax
    call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 120 0
```

- `asm("");` with `gcc -g -O0` produces:

```
.loc 1 119 0
```

- `asm("");` with `gcc -g -O1` produces:

```
.loc 1 119 0
```

- `asm("");` with `gcc -g -O3` produces:

```
.loc 1 119 0
```

- `asm("nop");` with `gcc -g -O0` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O1` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O3` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 120 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 120 0

```

- `i = 4`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 120 0

```

- `gi = 11`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 120 0

```

- `gi = 11`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 120 0

```

- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 120 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 120 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 120 0

```

- `f = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 120 0

```

- `gf = 12.0;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 120 0

```

- `gf = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 120 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 120 0

```

- `i++;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 120 0

```

- `i++;` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 120 0

```

- `i++;` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 120 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 120 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 120 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 120 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 120 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 120 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
    movslq    20(%rsp), %rax
    movl      a(,%rax,4), %eax
    movl      %eax, 24(%rsp)
    .loc 1 120 0

```

- `assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
    movl      -56(%rbp), %eax
    cmpl      $9, %eax
    jg        .L5
    .loc 1 119 0 is_stmt 0 discriminator 1
    movl      $119, %edx
    movl      $.LC4, %esi
    movl      $.LC5, %edi
    call      _I_default_handler
.L5:
    .loc 1 120 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
    movl      24(%rsp), %eax
    cmpl      $9, %eax
    jg        .L5
    .loc 1 119 0 is_stmt 0 discriminator 1
    movl      $119, %edx
    movl      $.LC4, %esi
    movl      $.LC5, %edi
    call      _I_default_handler
.LVL5:
.L5:
    .loc 1 120 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
    movl      24(%rsp), %eax
    cmpl      $9, %eax
    jle       .L274
.L5:
    .loc 1 120 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:


```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $119, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 120 0 is_stmt 1

```

- BSD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 120 0

```

- BSD_assert(i >= 10); with gcc -g -O3 produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 120 0

```

- TRAD_assert(i >= 10); with gcc -g -O0 produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
.LBB2:
        .loc 1 119 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $119, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
.LBE2:
        .loc 1 120 0 is_stmt 1

```

- `TRAD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB783:
.LBB784:
.LBB785:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $119, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE785:
.LBE784:
        .loc 1 119 0 discriminator 1
        movl    $1, %edi
        call    exit
.LVL7:
.L5:
.LBE783:
        .loc 1 120 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 120 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $119, %edx

```

```

        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 120 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 120 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 120 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 120 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 120 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 120 0

```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
        .loc 1 119 0
        movl    $1, -3132(%rbp)
        movq    $str, -2096(%rbp)
        jmp     .L5
.L8:
        .loc 1 119 0 is_stmt 0 discriminator 2
        call    __ctype_b_loc
        movq    (%rax), %rdx
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        movsbq  %al, %rax
        addq    %rax, %rax
        addq    %rdx, %rax
        movzwl  (%rax), %eax
        movzwl  %ax, %eax
        andl    $512, %eax
        testl   %eax, %eax
        jne     .L6
        .loc 1 119 0 discriminator 1
        movl    $0, -3132(%rbp)
        jmp     .L7
.L6:
        .loc 1 119 0 discriminator 2
        addq    $1, -2096(%rbp)
.L5:
        .loc 1 119 0 discriminator 1
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        testb   %al, %al
        jne     .L8
.L7:
        movl    -3132(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L9
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L9:

```

```
.loc 1 120 0 is_stmt 1
```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O1 produces:

```
.LVL5:
#NO_APP
.LBB269:
    .loc 1 119 0
    movzbl  str(%rip), %ebx
    testb   %bl, %bl
    je      .L5
    call     __ctype_b_loc

.LVL6:
    movq     (%rax), %rdx
    movl     $str, %eax

.LVL7:
.L7:
    .loc 1 119 0 is_stmt 0 discriminator 2
    movsbq   %bl, %rbx
    testb    $2, 1(%rdx,%rbx,2)
    je      .L6
    addq     $1, %rax

.LVL8:
    movzbl    (%rax), %ebx
    testb     %bl, %bl
    jne      .L7

.LVL9:
.L5:
.LBE269:
    .loc 1 120 0 is_stmt 1
```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O3 produces:

```
.LVL5:
#NO_APP
.LBB271:
    .loc 1 119 0
    movsbq   str(%rip), %rbx
    testb    %bl, %bl
    je      .L5
    call     __ctype_b_loc

.LVL6:
    movq     (%rax), %rdx
    movl     $str, %eax
    jmp      .L7

.LVL7:
    .p2align 4,,10
```

```

        .p2align 3
.L2074:
        .loc 1 119 0 is_stmt 0 discriminator 2
        addq    $1, %rax
.LVL8:
        movsbq  (%rax), %rbx
        testb   %b1, %b1
        je      .L5
.LVL9:
.L7:
        testb   $2, 1(%rdx,%rbx,2)
        jne     .L2074
.LVL10:
.LBE271:
        .loc 1 119 0
        movl    $119, %edx
        movl    $.LC11, %esi
        movl    $.LC12, %edi
        call    _I_default_handler
.LVL11:
.L5:
        .loc 1 120 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
        .loc 1 119 0
        movl    $1, -5188(%rbp)
        movl    $0, -5184(%rbp)
        jmp     .L5
.L11:
.LBB3:
        .loc 1 119 0 is_stmt 0 discriminator 2
        movq    $0, -2096(%rbp)
        movl    $0, -5180(%rbp)
        jmp     .L6
.L8:
        movl    -5184(%rbp), %eax
        cltq
        movl    a(,%rax,4), %edx
        movl    -5180(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        cmpl    %eax, %edx
        jne     .L7
        .loc 1 119 0 discriminator 1
        addq    $1, -2096(%rbp)
.L7:

```

```

        .loc 1 119 0 discriminator 2
        addl    $1, -5180(%rbp)
.L6:
        .loc 1 119 0 discriminator 1
        cmpl    $9, -5180(%rbp)
        jle     .L8
        .loc 1 119 0 discriminator 3
        movq    -2096(%rbp), %rax
.LBE3:
        cmpq    $1, %rax
        je      .L9
        .loc 1 119 0 discriminator 1
        movl    $0, -5188(%rbp)
        jmp     .L10
.L9:
        .loc 1 119 0 discriminator 2
        addl    $1, -5184(%rbp)
.L5:
        .loc 1 119 0 discriminator 1
        cmpl    $9, -5184(%rbp)
        jle     .L11
.L10:
        movl    -5188(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L12
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L12:
        .loc 1 120 0 is_stmt 1

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O3` produces:

```

        .loc 1 119 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```


- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O3` produces:

```

        .loc 1 119 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O3` produces:

```

        .loc 1 119 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 120 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 120 0

```

- `i = 4`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 120 0

```

- `gi = 11`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 120 0

```

- `gi = 11`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 120 0

```

- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 120 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 120 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 120 0

```

- `f = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 120 0

```

- `gf = 12.0;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 120 0

```

- `gf = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 120 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 120 0

```

- `i++;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 120 0

```

- `i++;` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 120 0

```

- `i++;` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 120 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 120 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 120 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 120 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 120 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    20(%rsp), %eax

```

```

    cltq
    movl    a(,%rax,4), %eax
    movl    %eax, 24(%rsp)
    .loc 1 120 0

```

- `j = a[i];` with `gcc -g -O3` produces:

```

    .loc 1 119 0
#NO_APP
    movslq  20(%rsp), %rax
    movl    a(,%rax,4), %eax
    movl    %eax, 24(%rsp)
    .loc 1 120 0

```

- `assert(i >= 10);` with `gcc -g -O0` produces:

```

    .loc 1 119 0
#NO_APP
    movl    -56(%rbp), %eax
    cmpl    $9, %eax
    jg      .L5
    .loc 1 119 0 is_stmt 0 discriminator 1
    movl    $119, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.L5:
    .loc 1 120 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

    .loc 1 119 0
#NO_APP
    movl    24(%rsp), %eax
    cmpl    $9, %eax
    jg      .L5
    .loc 1 119 0 is_stmt 0 discriminator 1
    movl    $119, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.LVL5:
.L5:
    .loc 1 120 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

    .loc 1 119 0
#NO_APP

```



```

        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 120 0

```

- BSD_assert(i >= 10); with gcc -g -O0 produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $119, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 120 0 is_stmt 1

```

- BSD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 120 0

```

- BSD_assert(i >= 10); with gcc -g -O3 produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 120 0

```

- TRAD_assert(i >= 10); with gcc -g -O0 produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
.LBB2:
        .loc 1 119 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $119, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi

```

```

        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
.LBE2:
        .loc 1 120 0 is_stmt 1

```

- `TRAD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB783:
.LBB784:
.LBB785:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $119, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE785:
.LBE784:
        .loc 1 119 0 discriminator 1
        movl    $1, %edi
        call    exit
.LVL7:
.L5:
.LBE783:
        .loc 1 120 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 120 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 120 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 120 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 120 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 120 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 120 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 120 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)));` with `gcc -g -O0` produces:

```

#NO_APP
.LBB2:
        .loc 1 119 0
        movl    $1, -3132(%rbp)
        movq    $str, -2096(%rbp)
        jmp     .L5
.L8:
        .loc 1 119 0 is_stmt 0 discriminator 2
        call    __ctype_b_loc
        movq    (%rax), %rdx
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        movsbq  %al, %rax
        addq    %rax, %rax
        addq    %rdx, %rax
        movzwl  (%rax), %eax
        movzwl  %ax, %eax
        andl    $512, %eax
        testl   %eax, %eax
        jne     .L6
        .loc 1 119 0 discriminator 1
        movl    $0, -3132(%rbp)
        jmp     .L7
.L6:
        .loc 1 119 0 discriminator 2
        addq    $1, -2096(%rbp)
.L5:
        .loc 1 119 0 discriminator 1
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        testb   %al, %al
        jne     .L8
.L7:
        movl    -3132(%rbp), %eax

```

```
.LBE2:
    testl    %eax, %eax
    jne      .L9
    movl     $119, %edx
    movl     $.LC4, %esi
    movl     $.LC5, %edi
    call     _I_default_handler
.L9:
    .loc 1 120 0 is_stmt 1
```

- `asm("");` with `gcc -g -O0` produces:

```
.loc 1 119 0
```

- `asm("");` with `gcc -g -O1` produces:

```
.loc 1 119 0
```

- `asm("");` with `gcc -g -O3` produces:

```
.loc 1 119 0
```

- `asm("nop");` with `gcc -g -O0` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O1` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O3` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```
.loc 1 119 0
# 119 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O3` produces:

```

        .loc 1 119 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 119 0

```

- `asm("");` with `gcc -g -O3` produces:

```

        .loc 1 119 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:


```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
# 119 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 120 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 120 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 120 0

```

- `gi = 11;` with `gcc -g -O0` produces:

```
        .loc 1 119 0
```

```
#NO_APP
```

```
    movl    $11, gi(%rip)
```

```
        .loc 1 120 0
```

- `gi = 11`; with `gcc -g -O1` produces:

```
        .loc 1 119 0
```

```
#NO_APP
```

```
    movl    $11, gi(%rip)
```

```
        .loc 1 120 0
```

- `gi = 11`; with `gcc -g -O3` produces:

```
        .loc 1 119 0
```

```
#NO_APP
```

```
    movl    $11, gi(%rip)
```

```
        .loc 1 120 0
```

- `f = 12.0`; with `gcc -g -O0` produces:

```
        .loc 1 119 0
```

```
#NO_APP
```

```
    movl    .LC4(%rip), %eax
```

```
    movl    %eax, -48(%rbp)
```

```
        .loc 1 120 0
```

- `f = 12.0`; with `gcc -g -O1` produces:

```
        .loc 1 119 0
```

```
#NO_APP
```

```
    movss   .LC4(%rip), %xmm0
```

```
    movss   %xmm0, 28(%rsp)
```

```
        .loc 1 120 0
```

- `f = 12.0`; with `gcc -g -O3` produces:

```
        .loc 1 119 0
```

```
#NO_APP
```

```
    movl    $0x41400000, 28(%rsp)
```

```
        .loc 1 120 0
```

- `gf = 12.0`; with `gcc -g -O0` produces:

```
        .loc 1 119 0
```

```
#NO_APP
```

```
    movl    .LC4(%rip), %eax
```

```
    movl    %eax, gf(%rip)
```

```
        .loc 1 120 0
```

- `gf = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movss    .LC4(%rip), %xmm0
        movss    %xmm0, gf(%rip)
        .loc 1 120 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl     $0x41400000, gf(%rip)
        .loc 1 120 0

```

- `i++;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl     -56(%rbp), %eax
        addl     $1, %eax
        movl     %eax, -56(%rbp)
        .loc 1 120 0

```

- `i++;` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl     24(%rsp), %eax
        addl     $1, %eax
        movl     %eax, 24(%rsp)
        .loc 1 120 0

```

- `i++;` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl     24(%rsp), %eax
        addl     $1, %eax
        movl     %eax, 24(%rsp)
        .loc 1 120 0

```

- `gi++;` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl     gi(%rip), %eax
        addl     $1, %eax
        movl     %eax, gi(%rip)
        .loc 1 120 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 120 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 120 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 120 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 120 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movslq  20(%rsp), %rax
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 120 0

```

- `assert(i >= 10)`; with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 120 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 120 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 120 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $119, %esi

```

```

        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 120 0 is_stmt 1

```

- BSD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 120 0

```

- BSD_assert(i >= 10); with gcc -g -O3 produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 120 0

```

- TRAD_assert(i >= 10); with gcc -g -O0 produces:

```

        .loc 1 119 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
.LBB2:
        .loc 1 119 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $119, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
.LBE2:
        .loc 1 120 0 is_stmt 1

```

- TRAD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5

```

```

.LVL5:
.LBB783:
.LBB784:
.LBB785:
    .loc 2 97 0 discriminator 1
    movl    $.LC4, %r9d
    movl    $119, %r8d
    movl    $.LC5, %ecx
    movl    $.LC6, %edx
    movl    $1, %esi
    movq    stderr(%rip), %rdi
    movl    $0, %eax
    call    __fprintf_chk
.LVL6:
.LBE785:
.LBE784:
    .loc 1 119 0 discriminator 1
    movl    $1, %edi
    call    exit
.LVL7:
.L5:
.LBE783:
    .loc 1 120 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

    .loc 1 119 0
#NO_APP
    movl    24(%rsp), %eax
    cmpl    $9, %eax
    jle     .L21
    .loc 1 120 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

    .loc 1 119 0
#NO_APP
    movl    -56(%rbp), %eax
    cmpl    $9, %eax
    jg      .L5
    .loc 1 119 0 is_stmt 0 discriminator 1
    movl    $119, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.L5:
    .loc 1 120 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:


```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 120 0 is_stmt 1

```

- I(i >= 10); with gcc -g -O3 produces:

```

        .loc 1 119 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 120 0

```

- DI(i >= 10); with gcc -g -O0 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 120 0

```

- DI(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 120 0

```

- DI(i >= 10); with gcc -g -O3 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 120 0

```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
    .loc 1 119 0
    movl    $1, -3132(%rbp)
    movq    $str, -2096(%rbp)
    jmp     .L5
.L8:
    .loc 1 119 0 is_stmt 0 discriminator 2
    call    __ctype_b_loc
    movq    (%rax), %rdx
    movq    -2096(%rbp), %rax
    movzbl  (%rax), %eax
    movsbq  %al, %rax
    addq    %rax, %rax
    addq    %rdx, %rax
    movzwl  (%rax), %eax
    movzwl  %ax, %eax
    andl    $512, %eax
    testl   %eax, %eax
    jne     .L6
    .loc 1 119 0 discriminator 1
    movl    $0, -3132(%rbp)
    jmp     .L7
.L6:
    .loc 1 119 0 discriminator 2
    addq    $1, -2096(%rbp)
.L5:
    .loc 1 119 0 discriminator 1
    movq    -2096(%rbp), %rax
    movzbl  (%rax), %eax
    testb   %al, %al
    jne     .L8
.L7:
    movl    -3132(%rbp), %eax
.LBE2:
    testl   %eax, %eax
    jne     .L9
    movl    $119, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.L9:
    .loc 1 120 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB269:

```

```

        .loc 1 119 0
        movzbl  str(%rip), %ebx
        testb  %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
.LVL7:
.L7:
        .loc 1 119 0 is_stmt 0 discriminator 2
        movsbq  %bl, %rbx
        testb   $2, 1(%rdx,%rbx,2)
        je      .L6
        addq    $1, %rax
.LVL8:
        movzbl  (%rax), %ebx
        testb   %bl, %bl
        jne     .L7
.LVL9:
.L5:
.LBE269:
        .loc 1 120 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O3` produces:

```

.LVL5:
#NO_APP
.LBB271:
        .loc 1 119 0
        movsbq  str(%rip), %rbx
        testb   %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
        jmp     .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 119 0 is_stmt 0 discriminator 2
        addq    $1, %rax
.LVL8:
        movsbq  (%rax), %rbx
        testb   %bl, %bl
        je      .L5
.LVL9:

```

```

.L7:
    testb    $2, 1(%rdx,%rbx,2)
    jne      .L2074
.LVL10:
.LBE271:
    .loc 1 119 0
    movl     $119, %edx
    movl     $.LC11, %esi
    movl     $.LC12, %edi
    call     _I_default_handler
.LVL11:
.L5:
    .loc 1 120 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
    .loc 1 119 0
    movl     $1, -5188(%rbp)
    movl     $0, -5184(%rbp)
    jmp      .L5
.L11:
.LBB3:
    .loc 1 119 0 is_stmt 0 discriminator 2
    movq     $0, -2096(%rbp)
    movl     $0, -5180(%rbp)
    jmp      .L6
.L8:
    movl     -5184(%rbp), %eax
    cltq
    movl     a(,%rax,4), %edx
    movl     -5180(%rbp), %eax
    cltq
    movl     a(,%rax,4), %eax
    cmpl     %eax, %edx
    jne      .L7
    .loc 1 119 0 discriminator 1
    addq     $1, -2096(%rbp)
.L7:
    .loc 1 119 0 discriminator 2
    addl     $1, -5180(%rbp)
.L6:
    .loc 1 119 0 discriminator 1
    cmpl     $9, -5180(%rbp)
    jle      .L8
    .loc 1 119 0 discriminator 3
    movq     -2096(%rbp), %rax
.LBE3:

```

```

        cmpq    $1, %rax
        je      .L9
        .loc 1 119 0 discriminator 1
        movl    $0, -5188(%rbp)
        jmp     .L10
.L9:
        .loc 1 119 0 discriminator 2
        addl    $1, -5184(%rbp)
.L5:
        .loc 1 119 0 discriminator 1
        cmpl    $9, -5184(%rbp)
        jle     .L11
.L10:
        movl    -5188(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L12
        movl    $119, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L12:
        .loc 1 120 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 119 0
        movl    $0, %ecx
.LBE526:
        .loc 1 118 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 119 0 discriminator 2
        movslq   %ecx, %rdi
        movslq   %eax, %rsi
        movl     a(,%rsi,4), %ebx

```

```

        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:
        addl    $1, %eax
.LVL10:
        cmpl    $9, %eax
        jle     .L5
.LBE527:
        .loc 1 119 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 119 0 discriminator 2
        addl    $1, %ecx
.LVL11:
        cmpl    $9, %ecx
        jle     .L1295
.LVL12:
.L1038:
.LBE528:
        .loc 1 120 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB528:
.LBB529:
        .loc 1 119 0
        movl    a(%rip), %r13d
        movl    a+4(%rip), %r12d
        movl    $a, %ecx
        movl    a+8(%rip), %ebp
        movl    a+12(%rip), %ebx
        movl    a+16(%rip), %r11d
        movl    a+20(%rip), %r10d
        movl    a+24(%rip), %r9d
        movl    a+28(%rip), %r8d
        movl    a+32(%rip), %edi
        movl    a+36(%rip), %esi
.LVL6:
.L5:
        movl    (%rcx), %edx
        xorl    %eax, %eax
        cmpl    %r13d, %edx
        sete    %al
.LVL7:
        xorl    %r14d, %r14d

```

```

        cmpl    %r12d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL8:
        xorl    %r14d, %r14d
        cmpl    %ebp, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL9:
        xorl    %r14d, %r14d
        cmpl    %ebx, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL10:
        xorl    %r14d, %r14d
        cmpl    %r11d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL11:
        xorl    %r14d, %r14d
        cmpl    %r10d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL12:
        xorl    %r14d, %r14d
        cmpl    %r9d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL13:
        xorl    %r14d, %r14d
        cmpl    %r8d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL14:
        xorl    %r14d, %r14d
        cmpl    %edi, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL15:
        cmpl    %esi, %edx
        sete    %dl
        movzbl  %dl, %edx
        addq    %rdx, %rax
.LVL16:
.LBE529:
        cmpq    $1, %rax
        jne     .L4164
        addq    $4, %rcx
        .loc 1 119 0 is_stmt 0 discriminator 2
        cmpq    $a+40, %rcx

```

```

        jne      .L5
.LVL17:
.L6:
.LBE528:
        .loc 1 120 0 is_stmt 1

```

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl     $.LC4, %edi
        movl     $0, %eax
        call     printf
        .loc 1 120 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 104 0
        movl     $.LC4, %esi
        movl     $1, %edi
        movl     $0, %eax
        call     __printf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 104 0
        movl     $.LC4, %esi
        movl     $1, %edi
        xorl     %eax, %eax
        call     __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 120 0

```

- `L("helloworldn");` with `gcc -g -O0` produces:


```

        .loc 1 119 0
#NO_APP
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 120 0

```

- L("helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- L("helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 120 0

```

- DL("helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 120 0

```

- `DL("helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 120 0

```

- `DL("helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 120 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 120 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi

```

```

        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 120 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 119 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 120 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1

```

```

        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 120 0

```

- `LHP(fprintf,log,"helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movq    -32(%rbp), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 120 0

```

- `LHP(fprintf,log,"helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    %rbx, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```
.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 97 0
    movl    $11, %edx
    movq    %rbx, %rcx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 120 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```
    .loc 1 119 0
#NO_APP
    movq    -40(%rbp), %rax
    movl    $.LC4, %esi
    movq    %rax, %rdi
    movl    $0, %eax
    call    L_buffer_printf
    .loc 1 120 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```
    .loc 1 119 0
#NO_APP
    movl    $.LC4, %esi
    movq    %rbx, %rdi
    movl    $0, %eax
    call    L_buffer_printf
.LVL6:
    .loc 1 120 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```
    .loc 1 119 0
#NO_APP
    movl    $.LC4, %esi
    movq    %rbx, %rdi
    xorl    %eax, %eax
    call    L_buffer_printf
.LVL279:
    .loc 1 120 0
```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 119 0
#NO_APP
        movl    $.LC4, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    syslog
        .loc 1 120 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        xorl    %eax, %eax
        call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 120 0

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```
.loc 1 120 0
```

- `asm("");` with `gcc -g -O3` produces:

```
.loc 1 120 0
```

- `asm("nop");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```


- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movslq  20(%rsp), %rax
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `assert(i >= 10)`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- `BSD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- `TRAD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1

```

```

        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O0` produces:

```

#NO_APP
.LBB2:
        .loc 1 120 0
        movl    $1, -3132(%rbp)
        movq    $str, -2096(%rbp)
        jmp     .L5
.L8:
        .loc 1 120 0 is_stmt 0 discriminator 2
        call    __ctype_b_loc
        movq    (%rax), %rdx
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        movsbq  %al, %rax
        addq    %rax, %rax
        addq    %rdx, %rax
        movzwl  (%rax), %eax
        movzwl  %ax, %eax

```



```

        andl    $512, %eax
        testl   %eax, %eax
        jne     .L6
        .loc 1 120 0 discriminator 1
        movl    $0, -3132(%rbp)
        jmp     .L7
.L6:
        .loc 1 120 0 discriminator 2
        addq    $1, -2096(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        testb   %al, %al
        jne     .L8
.L7:
        movl    -3132(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L9
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L9:
        .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB269:
        .loc 1 120 0
        movzbl  str(%rip), %ebx
        testb   %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
.LVL7:
.L7:
        .loc 1 120 0 is_stmt 0 discriminator 2
        movsbq   %bl, %rbx
        testb    $2, 1(%rdx,%rbx,2)
        je       .L6
        addq     $1, %rax
.LVL8:
        movzbl  (%rax), %ebx

```

```

        testb    %b1, %b1
        jne      .L7
.LVL9:
.L5:
.LBE269:
        .loc 1 121 0 is_stmt 1

```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB271:
        .loc 1 120 0
        movsbq   str(%rip), %rbx
        testb    %b1, %b1
        je       .L5
        call     __ctype_b_loc
.LVL6:
        movq     (%rax), %rdx
        movl     $str, %eax
        jmp      .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 120 0 is_stmt 0 discriminator 2
        addq     $1, %rax
.LVL8:
        movsbq   (%rax), %rbx
        testb    %b1, %b1
        je       .L5
.LVL9:
.L7:
        testb    $2, 1(%rdx,%rbx,2)
        jne      .L2074
.LVL10:
.LBE271:
        .loc 1 120 0
        movl     $120, %edx
        movl     $.LC11, %esi
        movl     $.LC12, %edi
        call     _I_default_handler
.LVL11:
.L5:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))); with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
    .loc 1 120 0
    movl    $1, -5188(%rbp)
    movl    $0, -5184(%rbp)
    jmp     .L5
.L11:
.LBB3:
    .loc 1 120 0 is_stmt 0 discriminator 2
    movq    $0, -2096(%rbp)
    movl    $0, -5180(%rbp)
    jmp     .L6
.L8:
    movl    -5184(%rbp), %eax
    cltq
    movl    a(,%rax,4), %edx
    movl    -5180(%rbp), %eax
    cltq
    movl    a(,%rax,4), %eax
    cmpl    %eax, %edx
    jne     .L7
    .loc 1 120 0 discriminator 1
    addq    $1, -2096(%rbp)
.L7:
    .loc 1 120 0 discriminator 2
    addl    $1, -5180(%rbp)
.L6:
    .loc 1 120 0 discriminator 1
    cmpl    $9, -5180(%rbp)
    jle     .L8
    .loc 1 120 0 discriminator 3
    movq    -2096(%rbp), %rax
.LBE3:
    cmpq    $1, %rax
    je      .L9
    .loc 1 120 0 discriminator 1
    movl    $0, -5188(%rbp)
    jmp     .L10
.L9:
    .loc 1 120 0 discriminator 2
    addl    $1, -5184(%rbp)
.L5:
    .loc 1 120 0 discriminator 1
    cmpl    $9, -5184(%rbp)
    jle     .L11
.L10:
    movl    -5188(%rbp), %eax
.LBE2:
    testl   %eax, %eax
    jne     .L12

```

```

        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L12:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 120 0
        movl    $0, %ecx
.LBE526:
        .loc 1 119 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 120 0 discriminator 2
        movslq  %ecx, %rdi
        movslq  %eax, %rsi
        movl    a(,%rsi,4), %ebx
        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:
        addl    $1, %eax
.LVL10:
        cmpl    $9, %eax
        jle     .L5
.LBE527:
        .loc 1 120 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 120 0 discriminator 2
        addl    $1, %ecx
.LVL11:
        cmpl    $9, %ecx

```

```

        jle      .L1295
.LVL12:
.L1038:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O3` produces:

```

        .loc 1 120 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 121 0

```


- `gf = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss    .LC4(%rip), %xmm0
        movss    %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl     $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl     -56(%rbp), %eax
        addl     $1, %eax
        movl     %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl     24(%rsp), %eax
        addl     $1, %eax
        movl     %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl     24(%rsp), %eax
        addl     $1, %eax
        movl     %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl     gi(%rip), %eax
        addl     $1, %eax
        movl     %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movslq  20(%rsp), %rax
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `assert(i >= 10)`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi

```

```

        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- BSD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- BSD_assert(i >= 10); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- TRAD_assert(i >= 10); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- TRAD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:

```

```

.LBB527:
    .loc 2 97 0 discriminator 1
    movl    $.LC4, %r9d
    movl    $120, %r8d
    movl    $.LC5, %ecx
    movl    $.LC6, %edx
    movl    $1, %esi
    movq    stderr(%rip), %rdi
    movl    $0, %eax
    call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
    .loc 1 120 0 discriminator 1
    movl    $1, %edi
    call    exit
.LVL7:
.L5:
    .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

    .loc 1 120 0
#NO_APP
    movl    24(%rsp), %eax
    cmpl    $9, %eax
    jle     .L21
    .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

    .loc 1 120 0
#NO_APP
    movl    -56(%rbp), %eax
    cmpl    $9, %eax
    jg      .L5
    .loc 1 120 0 is_stmt 0 discriminator 1
    movl    $120, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.L5:
    .loc 1 121 0 is_stmt 1

```

- `asm("");` with `gcc -g -O0` produces:

```

    .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```
.loc 1 120 0
```

- `asm("");` with `gcc -g -O3` produces:

```
.loc 1 120 0
```

- `asm("nop");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```


- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movslq  20(%rsp), %rax
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `assert(i >= 10)`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- `BSD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- `TRAD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1

```

```

        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O0` produces:

```

#NO_APP
.LBB2:
        .loc 1 120 0
        movl    $1, -3132(%rbp)
        movq    $str, -2096(%rbp)
        jmp     .L5
.L8:
        .loc 1 120 0 is_stmt 0 discriminator 2
        call    __ctype_b_loc
        movq    (%rax), %rdx
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        movsbq  %al, %rax
        addq    %rax, %rax
        addq    %rdx, %rax
        movzwl  (%rax), %eax
        movzwl  %ax, %eax

```

```

        andl    $512, %eax
        testl   %eax, %eax
        jne     .L6
        .loc 1 120 0 discriminator 1
        movl    $0, -3132(%rbp)
        jmp     .L7
.L6:
        .loc 1 120 0 discriminator 2
        addq    $1, -2096(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        testb   %al, %al
        jne     .L8
.L7:
        movl    -3132(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L9
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L9:
        .loc 1 121 0 is_stmt 1

```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB269:
        .loc 1 120 0
        movzbl  str(%rip), %ebx
        testb   %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
.LVL7:
.L7:
        .loc 1 120 0 is_stmt 0 discriminator 2
        movsbq   %bl, %rbx
        testb    $2, 1(%rdx,%rbx,2)
        je       .L6
        addq     $1, %rax
.LVL8:
        movzbl  (%rax), %ebx

```



```

        testb    %b1, %b1
        jne      .L7
.LVL9:
.L5:
.LBE269:
        .loc 1 121 0 is_stmt 1

```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB271:
        .loc 1 120 0
        movsbq   str(%rip), %rbx
        testb    %b1, %b1
        je       .L5
        call     __ctype_b_loc
.LVL6:
        movq     (%rax), %rdx
        movl     $str, %eax
        jmp      .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 120 0 is_stmt 0 discriminator 2
        addq     $1, %rax
.LVL8:
        movsbq   (%rax), %rbx
        testb    %b1, %b1
        je       .L5
.LVL9:
.L7:
        testb    $2, 1(%rdx,%rbx,2)
        jne      .L2074
.LVL10:
.LBE271:
        .loc 1 120 0
        movl     $120, %edx
        movl     $.LC10, %esi
        movl     $.LC11, %edi
        call     _I_default_handler
.LVL11:
.L5:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))); with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
    .loc 1 120 0
    movl    $1, -5188(%rbp)
    movl    $0, -5184(%rbp)
    jmp     .L5

.L11:
.LBB3:
    .loc 1 120 0 is_stmt 0 discriminator 2
    movq    $0, -2096(%rbp)
    movl    $0, -5180(%rbp)
    jmp     .L6

.L8:
    movl    -5184(%rbp), %eax
    cltq
    movl    a(,%rax,4), %edx
    movl    -5180(%rbp), %eax
    cltq
    movl    a(,%rax,4), %eax
    cmpl    %eax, %edx
    jne     .L7
    .loc 1 120 0 discriminator 1
    addq    $1, -2096(%rbp)

.L7:
    .loc 1 120 0 discriminator 2
    addl    $1, -5180(%rbp)

.L6:
    .loc 1 120 0 discriminator 1
    cmpl    $9, -5180(%rbp)
    jle     .L8
    .loc 1 120 0 discriminator 3
    movq    -2096(%rbp), %rax

.LBE3:
    cmpq    $1, %rax
    je      .L9
    .loc 1 120 0 discriminator 1
    movl    $0, -5188(%rbp)
    jmp     .L10

.L9:
    .loc 1 120 0 discriminator 2
    addl    $1, -5184(%rbp)

.L5:
    .loc 1 120 0 discriminator 1
    cmpl    $9, -5184(%rbp)
    jle     .L11

.L10:
    movl    -5188(%rbp), %eax

.LBE2:
    testl   %eax, %eax
    jne     .L12

```

```

        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L12:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 120 0
        movl    $0, %ecx
.LBE526:
        .loc 1 119 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 120 0 discriminator 2
        movslq  %ecx, %rdi
        movslq  %eax, %rsi
        movl    a(,%rsi,4), %ebx
        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:
        addl    $1, %eax
.LVL10:
        cmpl    $9, %eax
        jle     .L5
.LBE527:
        .loc 1 120 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 120 0 discriminator 2
        addl    $1, %ecx
.LVL11:
        cmpl    $9, %ecx

```

```

        jle      .L1295
.LVL12:
.L1038:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB528:
.LBB529:
        .loc 1 120 0
        movl    a(%rip), %r13d
        movl    a+4(%rip), %r12d
        movl    $a, %ecx
        movl    a+8(%rip), %ebp
        movl    a+12(%rip), %ebx
        movl    a+16(%rip), %r11d
        movl    a+20(%rip), %r10d
        movl    a+24(%rip), %r9d
        movl    a+28(%rip), %r8d
        movl    a+32(%rip), %edi
        movl    a+36(%rip), %esi
.LVL6:
.L5:
        movl    (%rcx), %edx
        xorl    %eax, %eax
        cmpl    %r13d, %edx
        sete    %al
.LVL7:
        xorl    %r14d, %r14d
        cmpl    %r12d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL8:
        xorl    %r14d, %r14d
        cmpl    %ebp, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL9:
        xorl    %r14d, %r14d
        cmpl    %ebx, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL10:
        xorl    %r14d, %r14d
        cmpl    %r11d, %edx
        sete    %r14b

```

```

        addq    %r14, %rax
.LVL11:
        xorl    %r14d, %r14d
        cmpl    %r10d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL12:
        xorl    %r14d, %r14d
        cmpl    %r9d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL13:
        xorl    %r14d, %r14d
        cmpl    %r8d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL14:
        xorl    %r14d, %r14d
        cmpl    %edi, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL15:
        cmpl    %esi, %edx
        sete    %dl
        movzbl  %dl, %edx
        addq    %rdx, %rax
.LVL16:
.LBE529:
        cmpq    $1, %rax
        jne     .L4164
        addq    $4, %rcx
        .loc 1 120 0 is_stmt 0 discriminator 2
        cmpq    $a+40, %rcx
        jne     .L5
.LVL17:
.L6:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %edi
        movl    $0, %eax
        call    printf
        .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    movl    $0, %eax
    call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
    .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    xorl    %eax, %eax
    call    __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O0` produces:

```

    .loc 1 120 0
#NO_APP
    movq    stdout(%rip), %rax
    movq    %rax, %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
    .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 97 0

```

```

        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```


- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al

```

```

        jne      .L1045
.L5:
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movq     -32(%rbp), %rax
        movq     %rax, %rcx
        movl     $11, %edx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq     %rbx, %rcx
        movl     $11, %edx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movl     $11, %edx
        movq     %rbx, %rcx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movq    -40(%rbp), %rax
        movl    $.LC4, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    L_buffer_printf
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        movl    $0, %eax
        call    L_buffer_printf
.LVL6:
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        xorl    %eax, %eax
        call    L_buffer_printf
.LVL279:
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    syslog
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:

```

```

        .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        xorl    %eax, %eax
        call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O3` produces:

```

        .loc 1 120 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- i = 4; with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- i = 4; with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- gi = 11; with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- gi = 11; with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- gi = 11; with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- f = 12.0; with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- f = 12.0; with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movss    .LC4(%rip), %xmm0
        movss    %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl     $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl     .LC4(%rip), %eax
        movl     %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss    .LC4(%rip), %xmm0
        movss    %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl     $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl     -56(%rbp), %eax
        addl     $1, %eax
        movl     %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O1` produces:


```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i];` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i];` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movslq  20(%rsp), %rax
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- `BSD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d

```

```

        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- `TRAD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1
        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)));` with `gcc -g -O0` produces:

```

#NO_APP
.LBB2:
        .loc 1 120 0
        movl    $1, -3132(%rbp)
        movq    $str, -2096(%rbp)
        jmp     .L5
.L8:
        .loc 1 120 0 is_stmt 0 discriminator 2
        call    __ctype_b_loc
        movq    (%rax), %rdx
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        movsbq  %al, %rax
        addq    %rax, %rax
        addq    %rdx, %rax
        movzwl  (%rax), %eax
        movzwl  %ax, %eax
        andl    $512, %eax
        testl   %eax, %eax
        jne     .L6
        .loc 1 120 0 discriminator 1
        movl    $0, -3132(%rbp)
        jmp     .L7
.L6:
        .loc 1 120 0 discriminator 2
        addq    $1, -2096(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        testb   %al, %al
        jne     .L8
.L7:
        movl    -3132(%rbp), %eax

```

```

.LBE2:
    testl    %eax, %eax
    jne      .L9
    movl     $120, %edx
    movl     $.LC4, %esi
    movl     $.LC5, %edi
    call     _I_default_handler
.L9:
    .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB269:
    .loc 1 120 0
    movzbl   str(%rip), %ebx
    testb    %bl, %bl
    je       .L5
    call     __ctype_b_loc
.LVL6:
    movq     (%rax), %rdx
    movl     $str, %eax
.LVL7:
.L7:
    .loc 1 120 0 is_stmt 0 discriminator 2
    movsbq   %bl, %rbx
    testb    $2, 1(%rdx,%rbx,2)
    je       .L6
    addq     $1, %rax
.LVL8:
    movzbl   (%rax), %ebx
    testb    %bl, %bl
    jne      .L7
.LVL9:
.L5:
.LBE269:
    .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O3` produces:

```

.LVL5:
#NO_APP
.LBB271:
    .loc 1 120 0
    movsbq   str(%rip), %rbx
    testb    %bl, %bl

```

```

        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
        jmp     .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 120 0 is_stmt 0 discriminator 2
        addq    $1, %rax
.LVL8:
        movsbq   (%rax), %rbx
        testb    %bl, %bl
        je      .L5
.LVL9:
.L7:
        testb    $2, 1(%rdx,%rbx,2)
        jne     .L2074
.LVL10:
.LBE271:
        .loc 1 120 0
        movl     $120, %edx
        movl     $.LC10, %esi
        movl     $.LC11, %edi
        call     _I_default_handler
.LVL11:
.L5:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
        .loc 1 120 0
        movl     $1, -5188(%rbp)
        movl     $0, -5184(%rbp)
        jmp     .L5
.L11:
.LBB3:
        .loc 1 120 0 is_stmt 0 discriminator 2
        movq     $0, -2096(%rbp)
        movl     $0, -5180(%rbp)
        jmp     .L6
.L8:
        movl     -5184(%rbp), %eax
        cltq
        movl     a(,%rax,4), %edx

```



```

        movl    -5180(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        cmpl    %eax, %edx
        jne     .L7
        .loc 1 120 0 discriminator 1
        addq     $1, -2096(%rbp)
.L7:
        .loc 1 120 0 discriminator 2
        addl     $1, -5180(%rbp)
.L6:
        .loc 1 120 0 discriminator 1
        cmpl     $9, -5180(%rbp)
        jle     .L8
        .loc 1 120 0 discriminator 3
        movq     -2096(%rbp), %rax
.LBE3:
        cmpq     $1, %rax
        je      .L9
        .loc 1 120 0 discriminator 1
        movl     $0, -5188(%rbp)
        jmp     .L10
.L9:
        .loc 1 120 0 discriminator 2
        addl     $1, -5184(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        cmpl     $9, -5184(%rbp)
        jle     .L11
.L10:
        movl     -5188(%rbp), %eax
.LBE2:
        testl    %eax, %eax
        jne     .L12
        movl     $120, %edx
        movl     $.LC4, %esi
        movl     $.LC5, %edi
        call     _I_default_handler
.L12:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 120 0
        movl     $0, %ecx
.LBE526:

```

```

        .loc 1 119 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 120 0 discriminator 2
        movslq  %ecx, %rdi
        movslq  %eax, %rsi
        movl    a(,%rsi,4), %ebx
        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:
        addl    $1, %eax
.LVL10:
        cmpl    $9, %eax
        jle     .L5
.LBE527:
        .loc 1 120 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 120 0 discriminator 2
        addl    $1, %ecx
.LVL11:
        cmpl    $9, %ecx
        jle     .L1295
.LVL12:
.L1038:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB528:
.LBB529:
        .loc 1 120 0
        movl    a(%rip), %r13d
        movl    a+4(%rip), %r12d

```

```

        movl    $a, %ecx
        movl    a+8(%rip), %ebp
        movl    a+12(%rip), %ebx
        movl    a+16(%rip), %r11d
        movl    a+20(%rip), %r10d
        movl    a+24(%rip), %r9d
        movl    a+28(%rip), %r8d
        movl    a+32(%rip), %edi
        movl    a+36(%rip), %esi
.LVL6:
.L5:
        movl    (%rcx), %edx
        xorl    %eax, %eax
        cmpl    %r13d, %edx
        sete    %al
.LVL7:
        xorl    %r14d, %r14d
        cmpl    %r12d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL8:
        xorl    %r14d, %r14d
        cmpl    %ebp, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL9:
        xorl    %r14d, %r14d
        cmpl    %ebx, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL10:
        xorl    %r14d, %r14d
        cmpl    %r11d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL11:
        xorl    %r14d, %r14d
        cmpl    %r10d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL12:
        xorl    %r14d, %r14d
        cmpl    %r9d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL13:
        xorl    %r14d, %r14d
        cmpl    %r8d, %edx
        sete    %r14b
        addq    %r14, %rax

```

```

.LVL14:
    xorl    %r14d, %r14d
    cmpl    %edi, %edx
    sete    %r14b
    addq    %r14, %rax
.LVL15:
    cmpl    %esi, %edx
    sete    %dl
    movzbl  %dl, %edx
    addq    %rdx, %rax
.LVL16:
.LBE529:
    cmpq    $1, %rax
    jne     .L4164
    addq    $4, %rcx
    .loc 1 120 0 is_stmt 0 discriminator 2
    cmpq    $a+40, %rcx
    jne     .L5
.LVL17:
.L6:
.LBE528:
    .loc 1 121 0 is_stmt 1

```

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

    .loc 1 120 0
#NO_APP
    movl    $.LC4, %edi
    movl    $0, %eax
    call    printf
    .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    movl    $0, %eax
    call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
    .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 104 0
        movl    $.LC4, %esi
        movl    $1, %edi
        xorl    %eax, %eax
        call    __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:

```

```

        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi

```

```

        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `LHP(fprintf,log,"helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movq    -32(%rbp), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 121 0

```


- LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:

```
.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 97 0
    movq    %rbx, %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL6:
.LBE527:
.LBE526:
    .loc 1 121 0
```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```
.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 97 0
    movl    $11, %edx
    movq    %rbx, %rcx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```
    .loc 1 120 0
#NO_APP
    movq    -40(%rbp), %rax
    movl    $.LC4, %esi
    movq    %rax, %rdi
    movl    $0, %eax
    call    L_buffer_printf
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```
    .loc 1 120 0
#NO_APP
```

```

        movl    $.LC4, %esi
        movq    %rbx, %rdi
        movl    $0, %eax
        call    L_buffer_printf
.LVL6:
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        xorl    %eax, %eax
        call    L_buffer_printf
.LVL279:
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    syslog
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 3 31 0
    movl    $.LC4, %edx
    movl    $1, %esi
    movl    $8, %edi
    xorl    %eax, %eax
    call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 121 0

```

- `asm("");` with `gcc -g -O0` produces:

```

    .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```

    .loc 1 120 0

```

- `asm("");` with `gcc -g -O3` produces:

```

    .loc 1 120 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

    .loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

    .loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

    .loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O3` produces:


```

        .loc 1 120 0
#NO_APP
    movslq    20(%rsp), %rax
    movl      a(,%rax,4), %eax
    movl      %eax, 24(%rsp)
    .loc 1 121 0

```

- `assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
    movl      -56(%rbp), %eax
    cmpl      $9, %eax
    jg        .L5
    .loc 1 120 0 is_stmt 0 discriminator 1
    movl      $120, %edx
    movl      $.LC4, %esi
    movl      $.LC5, %edi
    call      _I_default_handler
.L5:
    .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
    movl      24(%rsp), %eax
    cmpl      $9, %eax
    jg        .L5
    .loc 1 120 0 is_stmt 0 discriminator 1
    movl      $120, %edx
    movl      $.LC4, %esi
    movl      $.LC5, %edi
    call      _I_default_handler
.LVL5:
.L5:
    .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
    movl      24(%rsp), %eax
    cmpl      $9, %eax
    jle       .L274
.L5:
    .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- BSD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- BSD_assert(i >= 10); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- TRAD_assert(i >= 10); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- TRAD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1
        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O0` produces:

```

#NO_APP
.LBB2:
    .loc 1 120 0
    movl    $1, -3132(%rbp)
    movq    $str, -2096(%rbp)
    jmp     .L5
.L8:
    .loc 1 120 0 is_stmt 0 discriminator 2
    call    __ctype_b_loc
    movq    (%rax), %rdx
    movq    -2096(%rbp), %rax
    movzbl  (%rax), %eax
    movsbq  %al, %rax
    addq    %rax, %rax
    addq    %rdx, %rax
    movzwl  (%rax), %eax
    movzwl  %ax, %eax
    andl    $512, %eax
    testl   %eax, %eax
    jne     .L6
    .loc 1 120 0 discriminator 1
    movl    $0, -3132(%rbp)
    jmp     .L7
.L6:
    .loc 1 120 0 discriminator 2
    addq    $1, -2096(%rbp)
.L5:
    .loc 1 120 0 discriminator 1
    movq    -2096(%rbp), %rax
    movzbl  (%rax), %eax
    testb   %al, %al
    jne     .L8
.L7:
    movl    -3132(%rbp), %eax
.LBE2:
    testl   %eax, %eax
    jne     .L9
    movl    $120, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.L9:
    .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB269:

```

```

        .loc 1 120 0
        movzbl  str(%rip), %ebx
        testb   %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
.LVL7:
.L7:
        .loc 1 120 0 is_stmt 0 discriminator 2
        movsbq  %bl, %rbx
        testb   $2, 1(%rdx,%rbx,2)
        je      .L6
        addq    $1, %rax
.LVL8:
        movzbl  (%rax), %ebx
        testb   %bl, %bl
        jne     .L7
.LVL9:
.L5:
.LBE269:
        .loc 1 121 0 is_stmt 1

```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB271:
        .loc 1 120 0
        movsbq  str(%rip), %rbx
        testb   %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
        jmp     .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 120 0 is_stmt 0 discriminator 2
        addq    $1, %rax
.LVL8:
        movsbq  (%rax), %rbx
        testb   %bl, %bl
        je      .L5
.LVL9:

```

```

.L7:
    testb    $2, 1(%rdx,%rbx,2)
    jne      .L2074
.LVL10:
.LBE271:
    .loc 1 120 0
    movl     $120, %edx
    movl     $.LC10, %esi
    movl     $.LC11, %edi
    call     _I_default_handler
.LVL11:
.L5:
    .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
    .loc 1 120 0
    movl     $1, -5188(%rbp)
    movl     $0, -5184(%rbp)
    jmp      .L5
.L11:
.LBB3:
    .loc 1 120 0 is_stmt 0 discriminator 2
    movq     $0, -2096(%rbp)
    movl     $0, -5180(%rbp)
    jmp      .L6
.L8:
    movl     -5184(%rbp), %eax
    cltq
    movl     a(,%rax,4), %edx
    movl     -5180(%rbp), %eax
    cltq
    movl     a(,%rax,4), %eax
    cmpl     %eax, %edx
    jne      .L7
    .loc 1 120 0 discriminator 1
    addq     $1, -2096(%rbp)
.L7:
    .loc 1 120 0 discriminator 2
    addl     $1, -5180(%rbp)
.L6:
    .loc 1 120 0 discriminator 1
    cmpl     $9, -5180(%rbp)
    jle      .L8
    .loc 1 120 0 discriminator 3
    movq     -2096(%rbp), %rax
.LBE3:

```

```

        cmpq    $1, %rax
        je      .L9
        .loc 1 120 0 discriminator 1
        movl    $0, -5188(%rbp)
        jmp     .L10
.L9:
        .loc 1 120 0 discriminator 2
        addl    $1, -5184(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        cmpl    $9, -5184(%rbp)
        jle     .L11
.L10:
        movl    -5188(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L12
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L12:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 120 0
        movl    $0, %ecx
.LBE526:
        .loc 1 119 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 120 0 discriminator 2
        movslq   %ecx, %rdi
        movslq   %eax, %rsi
        movl     a(,%rsi,4), %ebx

```



```

        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:   addl    $1, %eax
.LVL10:  cmpl    $9, %eax
        jle     .L5
.LBE527: .loc 1 120 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 120 0 discriminator 2
        addl    $1, %ecx
.LVL11:  cmpl    $9, %ecx
        jle     .L1295
.LVL12:
.L1038:
.LBE528: .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB528:
.LBB529:
        .loc 1 120 0
        movl    a(%rip), %r13d
        movl    a+4(%rip), %r12d
        movl    $a, %ecx
        movl    a+8(%rip), %ebp
        movl    a+12(%rip), %ebx
        movl    a+16(%rip), %r11d
        movl    a+20(%rip), %r10d
        movl    a+24(%rip), %r9d
        movl    a+28(%rip), %r8d
        movl    a+32(%rip), %edi
        movl    a+36(%rip), %esi
.LVL6:
.L5:
        movl    (%rcx), %edx
        xorl    %eax, %eax
        cmpl    %r13d, %edx
        sete    %al
.LVL7:
        xorl    %r14d, %r14d

```

```

        cmpl    %r12d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL8:
        xorl    %r14d, %r14d
        cmpl    %ebp, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL9:
        xorl    %r14d, %r14d
        cmpl    %ebx, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL10:
        xorl    %r14d, %r14d
        cmpl    %r11d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL11:
        xorl    %r14d, %r14d
        cmpl    %r10d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL12:
        xorl    %r14d, %r14d
        cmpl    %r9d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL13:
        xorl    %r14d, %r14d
        cmpl    %r8d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL14:
        xorl    %r14d, %r14d
        cmpl    %edi, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL15:
        cmpl    %esi, %edx
        sete    %dl
        movzbl  %dl, %edx
        addq    %rdx, %rax
.LVL16:
.LBE529:
        cmpq    $1, %rax
        jne     .L4164
        addq    $4, %rcx
        .loc 1 120 0 is_stmt 0 discriminator 2
        cmpq    $a+40, %rcx

```

```

        jne      .L5
.LVL17:
.L6:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl     $.LC4, %edi
        movl     $0, %eax
        call     printf
        .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 104 0
        movl     $.LC4, %esi
        movl     $1, %edi
        movl     $0, %eax
        call     __printf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 104 0
        movl     $.LC4, %esi
        movl     $1, %edi
        xorl     %eax, %eax
        call     __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `DL("helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `DL("helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi

```

```

        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1

```

```

        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `LHP(fprintf,log,"helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movq    -32(%rbp), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 121 0

```

- `LHP(fprintf,log,"helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    %rbx, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```
.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 97 0
    movl    $11, %edx
    movq    %rbx, %rcx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```
    .loc 1 120 0
#NO_APP
    movq    -40(%rbp), %rax
    movl    $.LC4, %esi
    movq    %rax, %rdi
    movl    $0, %eax
    call    L_buffer_printf
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```
    .loc 1 120 0
#NO_APP
    movl    $.LC4, %esi
    movq    %rbx, %rdi
    movl    $0, %eax
    call    L_buffer_printf
.LVL6:
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```
    .loc 1 120 0
#NO_APP
    movl    $.LC4, %esi
    movq    %rbx, %rdi
    xorl    %eax, %eax
    call    L_buffer_printf
.LVL279:
    .loc 1 121 0
```


- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    syslog
        .loc 1 121 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        xorl    %eax, %eax
        call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```
.loc 1 120 0
```

- `asm("");` with `gcc -g -O3` produces:

```
.loc 1 120 0
```

- `asm("nop");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O3` produces:

```
.loc 1 120 0
```

- `asm("nop");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:


```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `f = 12.0;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- `f = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i];` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i];` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i];` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movslq  20(%rsp), %rax
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- `BSD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- `TRAD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1

```

```

        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O0` produces:

```

#NO_APP
.LBB2:
        .loc 1 120 0
        movl    $1, -3132(%rbp)
        movq    $str, -2096(%rbp)
        jmp     .L5
.L8:
        .loc 1 120 0 is_stmt 0 discriminator 2
        call    __ctype_b_loc
        movq    (%rax), %rdx
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        movsbq  %al, %rax
        addq    %rax, %rax
        addq    %rdx, %rax
        movzwl  (%rax), %eax
        movzwl  %ax, %eax

```



```

        andl    $512, %eax
        testl   %eax, %eax
        jne     .L6
        .loc 1 120 0 discriminator 1
        movl    $0, -3132(%rbp)
        jmp     .L7
.L6:
        .loc 1 120 0 discriminator 2
        addq    $1, -2096(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        testb   %al, %al
        jne     .L8
.L7:
        movl    -3132(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L9
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L9:
        .loc 1 121 0 is_stmt 1

```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB269:
        .loc 1 120 0
        movzbl  str(%rip), %ebx
        testb   %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
.LVL7:
.L7:
        .loc 1 120 0 is_stmt 0 discriminator 2
        movsbq   %bl, %rbx
        testb    $2, 1(%rdx,%rbx,2)
        je       .L6
        addq     $1, %rax
.LVL8:
        movzbl  (%rax), %ebx

```

```

        testb    %b1, %b1
        jne      .L7
.LVL9:
.L5:
.LBE269:
        .loc 1 121 0 is_stmt 1

```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB271:
        .loc 1 120 0
        movsbq   str(%rip), %rbx
        testb    %b1, %b1
        je       .L5
        call     __ctype_b_loc
.LVL6:
        movq     (%rax), %rdx
        movl     $str, %eax
        jmp      .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 120 0 is_stmt 0 discriminator 2
        addq     $1, %rax
.LVL8:
        movsbq   (%rax), %rbx
        testb    %b1, %b1
        je       .L5
.LVL9:
.L7:
        testb    $2, 1(%rdx,%rbx,2)
        jne      .L2074
.LVL10:
.LBE271:
        .loc 1 120 0
        movl     $120, %edx
        movl     $.LC10, %esi
        movl     $.LC11, %edi
        call     _I_default_handler
.LVL11:
.L5:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))); with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
    .loc 1 120 0
    movl    $1, -5188(%rbp)
    movl    $0, -5184(%rbp)
    jmp     .L5
.L11:
.LBB3:
    .loc 1 120 0 is_stmt 0 discriminator 2
    movq    $0, -2096(%rbp)
    movl    $0, -5180(%rbp)
    jmp     .L6
.L8:
    movl    -5184(%rbp), %eax
    cltq
    movl    a(,%rax,4), %edx
    movl    -5180(%rbp), %eax
    cltq
    movl    a(,%rax,4), %eax
    cmpl    %eax, %edx
    jne     .L7
    .loc 1 120 0 discriminator 1
    addq    $1, -2096(%rbp)
.L7:
    .loc 1 120 0 discriminator 2
    addl    $1, -5180(%rbp)
.L6:
    .loc 1 120 0 discriminator 1
    cmpl    $9, -5180(%rbp)
    jle     .L8
    .loc 1 120 0 discriminator 3
    movq    -2096(%rbp), %rax
.LBE3:
    cmpq    $1, %rax
    je      .L9
    .loc 1 120 0 discriminator 1
    movl    $0, -5188(%rbp)
    jmp     .L10
.L9:
    .loc 1 120 0 discriminator 2
    addl    $1, -5184(%rbp)
.L5:
    .loc 1 120 0 discriminator 1
    cmpl    $9, -5184(%rbp)
    jle     .L11
.L10:
    movl    -5188(%rbp), %eax
.LBE2:
    testl   %eax, %eax
    jne     .L12

```

```

        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L12:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 120 0
        movl    $0, %ecx
.LBE526:
        .loc 1 119 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 120 0 discriminator 2
        movslq  %ecx, %rdi
        movslq  %eax, %rsi
        movl    a(,%rsi,4), %ebx
        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:
        addl    $1, %eax
.LVL10:
        cmpl    $9, %eax
        jle     .L5
.LBE527:
        .loc 1 120 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 120 0 discriminator 2
        addl    $1, %ecx
.LVL11:
        cmpl    $9, %ecx

```

```

        jle      .L1295
.LVL12:
.L1038:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB528:
.LBB529:
        .loc 1 120 0
        movl    a(%rip), %r13d
        movl    a+4(%rip), %r12d
        movl    $a, %ecx
        movl    a+8(%rip), %ebp
        movl    a+12(%rip), %ebx
        movl    a+16(%rip), %r11d
        movl    a+20(%rip), %r10d
        movl    a+24(%rip), %r9d
        movl    a+28(%rip), %r8d
        movl    a+32(%rip), %edi
        movl    a+36(%rip), %esi
.LVL6:
.L5:
        movl    (%rcx), %edx
        xorl    %eax, %eax
        cmpl    %r13d, %edx
        sete    %al
.LVL7:
        xorl    %r14d, %r14d
        cmpl    %r12d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL8:
        xorl    %r14d, %r14d
        cmpl    %ebp, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL9:
        xorl    %r14d, %r14d
        cmpl    %ebx, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL10:
        xorl    %r14d, %r14d
        cmpl    %r11d, %edx
        sete    %r14b

```

```

        addq    %r14, %rax
.LVL11:
        xorl    %r14d, %r14d
        cmpl    %r10d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL12:
        xorl    %r14d, %r14d
        cmpl    %r9d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL13:
        xorl    %r14d, %r14d
        cmpl    %r8d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL14:
        xorl    %r14d, %r14d
        cmpl    %edi, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL15:
        cmpl    %esi, %edx
        sete    %dl
        movzbl  %dl, %edx
        addq    %rdx, %rax
.LVL16:
.LBE529:
        cmpq    $1, %rax
        jne     .L4164
        addq    $4, %rcx
        .loc 1 120 0 is_stmt 0 discriminator 2
        cmpq    $a+40, %rcx
        jne     .L5
.LVL17:
.L6:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %edi
        movl    $0, %eax
        call    printf
        .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    movl    $0, %eax
    call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
    .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    xorl    %eax, %eax
    call    __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O0` produces:

```

    .loc 1 120 0
#NO_APP
    movq    stdout(%rip), %rax
    movq    %rax, %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
    .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 97 0

```

```

        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O0 produces:


```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al

```

```

        jne      .L1045
.L5:
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movq     -32(%rbp), %rax
        movq     %rax, %rcx
        movl     $11, %edx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq     %rbx, %rcx
        movl     $11, %edx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movl     $11, %edx
        movq     %rbx, %rcx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movq    -40(%rbp), %rax
        movl    $.LC4, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    L_buffer_printf
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        movl    $0, %eax
        call    L_buffer_printf
.LVL6:
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        xorl    %eax, %eax
        call    L_buffer_printf
.LVL279:
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    syslog
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:

```

```

        .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        xorl    %eax, %eax
        call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O3` produces:

```

        .loc 1 120 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- i = 4; with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- i = 4; with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- gi = 11; with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- gi = 11; with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- gi = 11; with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- f = 12.0; with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- f = 12.0; with gcc -g -O1 produces:


```

        .loc 1 120 0
#NO_APP
        movss    .LC4(%rip), %xmm0
        movss    %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl     $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl     .LC4(%rip), %eax
        movl     %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss    .LC4(%rip), %xmm0
        movss    %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl     $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl     -56(%rbp), %eax
        addl     $1, %eax
        movl     %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movslq  20(%rsp), %rax
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `assert(i >= 10)`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10)`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- `BSD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d

```

```

        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- `TRAD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1
        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O0` produces:

```

#NO_APP
.LBB2:
        .loc 1 120 0
        movl    $1, -3132(%rbp)
        movq    $str, -2096(%rbp)
        jmp     .L5
.L8:
        .loc 1 120 0 is_stmt 0 discriminator 2
        call    __ctype_b_loc
        movq    (%rax), %rdx
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        movsbq  %al, %rax
        addq    %rax, %rax
        addq    %rdx, %rax
        movzwl  (%rax), %eax
        movzwl  %ax, %eax
        andl    $512, %eax
        testl   %eax, %eax
        jne     .L6
        .loc 1 120 0 discriminator 1
        movl    $0, -3132(%rbp)
        jmp     .L7
.L6:
        .loc 1 120 0 discriminator 2
        addq    $1, -2096(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        testb   %al, %al
        jne     .L8
.L7:
        movl    -3132(%rbp), %eax

```

```

.LBE2:
    testl    %eax, %eax
    jne      .L9
    movl     $120, %edx
    movl     $.LC4, %esi
    movl     $.LC5, %edi
    call     _I_default_handler
.L9:
    .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB269:
    .loc 1 120 0
    movzbl   str(%rip), %ebx
    testb    %bl, %bl
    je       .L5
    call     __ctype_b_loc
.LVL6:
    movq     (%rax), %rdx
    movl     $str, %eax
.LVL7:
.L7:
    .loc 1 120 0 is_stmt 0 discriminator 2
    movsbq   %bl, %rbx
    testb    $2, 1(%rdx,%rbx,2)
    je       .L6
    addq     $1, %rax
.LVL8:
    movzbl   (%rax), %ebx
    testb    %bl, %bl
    jne      .L7
.LVL9:
.L5:
.LBE269:
    .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O3` produces:

```

.LVL5:
#NO_APP
.LBB271:
    .loc 1 120 0
    movsbq   str(%rip), %rbx
    testb    %bl, %bl

```



```

        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
        jmp     .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 120 0 is_stmt 0 discriminator 2
        addq    $1, %rax
.LVL8:
        movsbq   (%rax), %rbx
        testb    %bl, %bl
        je      .L5
.LVL9:
.L7:
        testb    $2, 1(%rdx,%rbx,2)
        jne     .L2074
.LVL10:
.LBE271:
        .loc 1 120 0
        movl     $120, %edx
        movl     $.LC10, %esi
        movl     $.LC11, %edi
        call     _I_default_handler
.LVL11:
.L5:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
        .loc 1 120 0
        movl     $1, -5188(%rbp)
        movl     $0, -5184(%rbp)
        jmp     .L5
.L11:
.LBB3:
        .loc 1 120 0 is_stmt 0 discriminator 2
        movq     $0, -2096(%rbp)
        movl     $0, -5180(%rbp)
        jmp     .L6
.L8:
        movl     -5184(%rbp), %eax
        cltq
        movl     a(,%rax,4), %edx

```

```

        movl    -5180(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        cmpl    %eax, %edx
        jne     .L7
        .loc 1 120 0 discriminator 1
        addq     $1, -2096(%rbp)
.L7:
        .loc 1 120 0 discriminator 2
        addl     $1, -5180(%rbp)
.L6:
        .loc 1 120 0 discriminator 1
        cmpl     $9, -5180(%rbp)
        jle     .L8
        .loc 1 120 0 discriminator 3
        movq     -2096(%rbp), %rax
.LBE3:
        cmpq     $1, %rax
        je      .L9
        .loc 1 120 0 discriminator 1
        movl     $0, -5188(%rbp)
        jmp     .L10
.L9:
        .loc 1 120 0 discriminator 2
        addl     $1, -5184(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        cmpl     $9, -5184(%rbp)
        jle     .L11
.L10:
        movl     -5188(%rbp), %eax
.LBE2:
        testl    %eax, %eax
        jne     .L12
        movl     $120, %edx
        movl     $.LC4, %esi
        movl     $.LC5, %edi
        call     _I_default_handler
.L12:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 120 0
        movl     $0, %ecx
.LBE526:

```

```

        .loc 1 119 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 120 0 discriminator 2
        movslq  %ecx, %rdi
        movslq  %eax, %rsi
        movl    a(,%rsi,4), %ebx
        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:
        addl    $1, %eax
.LVL10:
        cmpl    $9, %eax
        jle     .L5
.LBE527:
        .loc 1 120 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 120 0 discriminator 2
        addl    $1, %ecx
.LVL11:
        cmpl    $9, %ecx
        jle     .L1295
.LVL12:
.L1038:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB528:
.LBB529:
        .loc 1 120 0
        movl    a(%rip), %r13d
        movl    a+4(%rip), %r12d

```

```

        movl    $a, %ecx
        movl    a+8(%rip), %ebp
        movl    a+12(%rip), %ebx
        movl    a+16(%rip), %r11d
        movl    a+20(%rip), %r10d
        movl    a+24(%rip), %r9d
        movl    a+28(%rip), %r8d
        movl    a+32(%rip), %edi
        movl    a+36(%rip), %esi
.LVL6:
.L5:
        movl    (%rcx), %edx
        xorl    %eax, %eax
        cmpl    %r13d, %edx
        sete    %al
.LVL7:
        xorl    %r14d, %r14d
        cmpl    %r12d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL8:
        xorl    %r14d, %r14d
        cmpl    %ebp, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL9:
        xorl    %r14d, %r14d
        cmpl    %ebx, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL10:
        xorl    %r14d, %r14d
        cmpl    %r11d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL11:
        xorl    %r14d, %r14d
        cmpl    %r10d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL12:
        xorl    %r14d, %r14d
        cmpl    %r9d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL13:
        xorl    %r14d, %r14d
        cmpl    %r8d, %edx
        sete    %r14b
        addq    %r14, %rax

```

```

.LVL14:
    xorl    %r14d, %r14d
    cmpl    %edi, %edx
    sete    %r14b
    addq    %r14, %rax
.LVL15:
    cmpl    %esi, %edx
    sete    %dl
    movzbl  %dl, %edx
    addq    %rdx, %rax
.LVL16:
.LBE529:
    cmpq    $1, %rax
    jne     .L4164
    addq    $4, %rcx
    .loc 1 120 0 is_stmt 0 discriminator 2
    cmpq    $a+40, %rcx
    jne     .L5
.LVL17:
.L6:
.LBE528:
    .loc 1 121 0 is_stmt 1

```

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

    .loc 1 120 0
#NO_APP
    movl    $.LC4, %edi
    movl    $0, %eax
    call    printf
    .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    movl    $0, %eax
    call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
    .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 104 0
        movl    $.LC4, %esi
        movl    $1, %edi
        xorl    %eax, %eax
        call    __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:

```

```

        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi

```



```

        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `LHP(fprintf,log,"helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movq    -32(%rbp), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:

```
.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 97 0
    movq    %rbx, %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL6:
.LBE527:
.LBE526:
    .loc 1 121 0
```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```
.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 97 0
    movl    $11, %edx
    movq    %rbx, %rcx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```
    .loc 1 120 0
#NO_APP
    movq    -40(%rbp), %rax
    movl    $.LC4, %esi
    movq    %rax, %rdi
    movl    $0, %eax
    call    L_buffer_printf
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```
    .loc 1 120 0
#NO_APP
```

```

        movl    $.LC4, %esi
        movq    %rbx, %rdi
        movl    $0, %eax
        call    L_buffer_printf
.LVL6:
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        xorl    %eax, %eax
        call    L_buffer_printf
.LVL279:
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    syslog
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 3 31 0
    movl    $.LC4, %edx
    movl    $1, %esi
    movl    $8, %edi
    xorl    %eax, %eax
    call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 121 0

```

- `asm("");` with `gcc -g -O0` produces:

```

    .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```

    .loc 1 120 0

```

- `asm("");` with `gcc -g -O3` produces:

```

    .loc 1 120 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

    .loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

    .loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

    .loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```


- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
    movslq    20(%rsp), %rax
    movl      a(,%rax,4), %eax
    movl      %eax, 24(%rsp)
    .loc 1 121 0

```

- `assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
    movl      -56(%rbp), %eax
    cmpl      $9, %eax
    jg        .L5
    .loc 1 120 0 is_stmt 0 discriminator 1
    movl      $120, %edx
    movl      $.LC4, %esi
    movl      $.LC5, %edi
    call      _I_default_handler
.L5:
    .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
    movl      24(%rsp), %eax
    cmpl      $9, %eax
    jg        .L5
    .loc 1 120 0 is_stmt 0 discriminator 1
    movl      $120, %edx
    movl      $.LC4, %esi
    movl      $.LC5, %edi
    call      _I_default_handler
.LVL5:
.L5:
    .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
    movl      24(%rsp), %eax
    cmpl      $9, %eax
    jle       .L274
.L5:
    .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- BSD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- BSD_assert(i >= 10); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- TRAD_assert(i >= 10); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- TRAD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1
        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O0` produces:

```

#NO_APP
.LBB2:
    .loc 1 120 0
    movl    $1, -3132(%rbp)
    movq    $str, -2096(%rbp)
    jmp     .L5
.L8:
    .loc 1 120 0 is_stmt 0 discriminator 2
    call    __ctype_b_loc
    movq    (%rax), %rdx
    movq    -2096(%rbp), %rax
    movzbl  (%rax), %eax
    movsbq  %al, %rax
    addq    %rax, %rax
    addq    %rdx, %rax
    movzwl  (%rax), %eax
    movzwl  %ax, %eax
    andl    $512, %eax
    testl   %eax, %eax
    jne     .L6
    .loc 1 120 0 discriminator 1
    movl    $0, -3132(%rbp)
    jmp     .L7
.L6:
    .loc 1 120 0 discriminator 2
    addq    $1, -2096(%rbp)
.L5:
    .loc 1 120 0 discriminator 1
    movq    -2096(%rbp), %rax
    movzbl  (%rax), %eax
    testb   %al, %al
    jne     .L8
.L7:
    movl    -3132(%rbp), %eax
.LBE2:
    testl   %eax, %eax
    jne     .L9
    movl    $120, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.L9:
    .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB269:

```

```

        .loc 1 120 0
        movzbl  str(%rip), %ebx
        testb  %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
.LVL7:
.L7:
        .loc 1 120 0 is_stmt 0 discriminator 2
        movsbq  %bl, %rbx
        testb   $2, 1(%rdx,%rbx,2)
        je      .L6
        addq    $1, %rax
.LVL8:
        movzbl  (%rax), %ebx
        testb   %bl, %bl
        jne     .L7
.LVL9:
.L5:
.LBE269:
        .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O3` produces:

```

.LVL5:
#NO_APP
.LBB271:
        .loc 1 120 0
        movsbq  str(%rip), %rbx
        testb   %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
        jmp     .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 120 0 is_stmt 0 discriminator 2
        addq    $1, %rax
.LVL8:
        movsbq  (%rax), %rbx
        testb   %bl, %bl
        je      .L5
.LVL9:

```

```

.L7:
    testb    $2, 1(%rdx,%rbx,2)
    jne      .L2074
.LVL10:
.LBE271:
    .loc 1 120 0
    movl     $120, %edx
    movl     $.LC10, %esi
    movl     $.LC11, %edi
    call     _I_default_handler
.LVL11:
.L5:
    .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
    .loc 1 120 0
    movl     $1, -5188(%rbp)
    movl     $0, -5184(%rbp)
    jmp      .L5
.L11:
.LBB3:
    .loc 1 120 0 is_stmt 0 discriminator 2
    movq     $0, -2096(%rbp)
    movl     $0, -5180(%rbp)
    jmp      .L6
.L8:
    movl     -5184(%rbp), %eax
    cltq
    movl     a(,%rax,4), %edx
    movl     -5180(%rbp), %eax
    cltq
    movl     a(,%rax,4), %eax
    cmpl     %eax, %edx
    jne      .L7
    .loc 1 120 0 discriminator 1
    addq     $1, -2096(%rbp)
.L7:
    .loc 1 120 0 discriminator 2
    addl     $1, -5180(%rbp)
.L6:
    .loc 1 120 0 discriminator 1
    cmpl     $9, -5180(%rbp)
    jle      .L8
    .loc 1 120 0 discriminator 3
    movq     -2096(%rbp), %rax
.LBE3:

```



```

        cmpq    $1, %rax
        je      .L9
        .loc 1 120 0 discriminator 1
        movl    $0, -5188(%rbp)
        jmp     .L10
.L9:
        .loc 1 120 0 discriminator 2
        addl    $1, -5184(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        cmpl    $9, -5184(%rbp)
        jle     .L11
.L10:
        movl    -5188(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L12
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L12:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 120 0
        movl    $0, %ecx
.LBE526:
        .loc 1 119 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 120 0 discriminator 2
        movslq   %ecx, %rdi
        movslq   %eax, %rsi
        movl     a(,%rsi,4), %ebx

```

```

        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:   addl    $1, %eax
.LVL10:  cmpl    $9, %eax
        jle     .L5
.LBE527:
        .loc 1 120 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 120 0 discriminator 2
        addl    $1, %ecx
.LVL11:  cmpl    $9, %ecx
        jle     .L1295
.LVL12:
.L1038:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB528:
.LBB529:
        .loc 1 120 0
        movl    a(%rip), %r13d
        movl    a+4(%rip), %r12d
        movl    $a, %ecx
        movl    a+8(%rip), %ebp
        movl    a+12(%rip), %ebx
        movl    a+16(%rip), %r11d
        movl    a+20(%rip), %r10d
        movl    a+24(%rip), %r9d
        movl    a+28(%rip), %r8d
        movl    a+32(%rip), %edi
        movl    a+36(%rip), %esi
.LVL6:
.L5:
        movl    (%rcx), %edx
        xorl    %eax, %eax
        cmpl    %r13d, %edx
        sete    %al
.LVL7:
        xorl    %r14d, %r14d

```

```

        cmpl    %r12d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL8:
        xorl    %r14d, %r14d
        cmpl    %ebp, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL9:
        xorl    %r14d, %r14d
        cmpl    %ebx, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL10:
        xorl    %r14d, %r14d
        cmpl    %r11d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL11:
        xorl    %r14d, %r14d
        cmpl    %r10d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL12:
        xorl    %r14d, %r14d
        cmpl    %r9d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL13:
        xorl    %r14d, %r14d
        cmpl    %r8d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL14:
        xorl    %r14d, %r14d
        cmpl    %edi, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL15:
        cmpl    %esi, %edx
        sete    %dl
        movzbl  %dl, %edx
        addq    %rdx, %rax
.LVL16:
.LBE529:
        cmpq    $1, %rax
        jne     .L4164
        addq    $4, %rcx
        .loc 1 120 0 is_stmt 0 discriminator 2
        cmpq    $a+40, %rcx

```

```

        jne      .L5
.LVL17:
.L6:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl     $.LC4, %edi
        movl     $0, %eax
        call     printf
        .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 104 0
        movl     $.LC4, %esi
        movl     $1, %edi
        movl     $0, %eax
        call     __printf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 104 0
        movl     $.LC4, %esi
        movl     $1, %edi
        xorl     %eax, %eax
        call     __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `DL("helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `DL("helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi

```

```

        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1

```

```

        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `LHP(fprintf,log,"helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movq    -32(%rbp), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 121 0

```

- `LHP(fprintf,log,"helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    %rbx, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```


- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```
.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 97 0
    movl    $11, %edx
    movq    %rbx, %rcx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```
    .loc 1 120 0
#NO_APP
    movq    -40(%rbp), %rax
    movl    $.LC4, %esi
    movq    %rax, %rdi
    movl    $0, %eax
    call    L_buffer_printf
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```
    .loc 1 120 0
#NO_APP
    movl    $.LC4, %esi
    movq    %rbx, %rdi
    movl    $0, %eax
    call    L_buffer_printf
.LVL6:
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```
    .loc 1 120 0
#NO_APP
    movl    $.LC4, %esi
    movq    %rbx, %rdi
    xorl    %eax, %eax
    call    L_buffer_printf
.LVL279:
    .loc 1 121 0
```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    syslog
        .loc 1 121 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        xorl    %eax, %eax
        call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```
.loc 1 120 0
```

- `asm("");` with `gcc -g -O3` produces:

```
.loc 1 120 0
```

- `asm("nop");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movslq  20(%rsp), %rax
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `assert(i >= 10)`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```


- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- `BSD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- `TRAD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1

```

```

        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O0` produces:

```

#NO_APP
.LBB2:
        .loc 1 120 0
        movl    $1, -3132(%rbp)
        movq    $str, -2096(%rbp)
        jmp     .L5
.L8:
        .loc 1 120 0 is_stmt 0 discriminator 2
        call    __ctype_b_loc
        movq    (%rax), %rdx
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        movsbq  %al, %rax
        addq    %rax, %rax
        addq    %rdx, %rax
        movzwl  (%rax), %eax
        movzwl  %ax, %eax

```

```

        andl    $512, %eax
        testl   %eax, %eax
        jne     .L6
        .loc 1 120 0 discriminator 1
        movl    $0, -3132(%rbp)
        jmp     .L7
.L6:
        .loc 1 120 0 discriminator 2
        addq    $1, -2096(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        testb   %al, %al
        jne     .L8
.L7:
        movl    -3132(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L9
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L9:
        .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB269:
        .loc 1 120 0
        movzbl  str(%rip), %ebx
        testb   %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
.LVL7:
.L7:
        .loc 1 120 0 is_stmt 0 discriminator 2
        movsbq   %bl, %rbx
        testb    $2, 1(%rdx,%rbx,2)
        je       .L6
        addq     $1, %rax
.LVL8:
        movzbl  (%rax), %ebx

```

```

        testb    %b1, %b1
        jne      .L7
.LVL9:
.L5:
.LBE269:
        .loc 1 121 0 is_stmt 1

```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB271:
        .loc 1 120 0
        movsbq   str(%rip), %rbx
        testb    %b1, %b1
        je       .L5
        call     __ctype_b_loc
.LVL6:
        movq     (%rax), %rdx
        movl     $str, %eax
        jmp      .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 120 0 is_stmt 0 discriminator 2
        addq     $1, %rax
.LVL8:
        movsbq   (%rax), %rbx
        testb    %b1, %b1
        je       .L5
.LVL9:
.L7:
        testb    $2, 1(%rdx,%rbx,2)
        jne      .L2074
.LVL10:
.LBE271:
        .loc 1 120 0
        movl     $120, %edx
        movl     $.LC10, %esi
        movl     $.LC11, %edi
        call     _I_default_handler
.LVL11:
.L5:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))); with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
    .loc 1 120 0
    movl    $1, -5188(%rbp)
    movl    $0, -5184(%rbp)
    jmp     .L5
.L11:
.LBB3:
    .loc 1 120 0 is_stmt 0 discriminator 2
    movq    $0, -2096(%rbp)
    movl    $0, -5180(%rbp)
    jmp     .L6
.L8:
    movl    -5184(%rbp), %eax
    cltq
    movl    a(,%rax,4), %edx
    movl    -5180(%rbp), %eax
    cltq
    movl    a(,%rax,4), %eax
    cmpl    %eax, %edx
    jne     .L7
    .loc 1 120 0 discriminator 1
    addq    $1, -2096(%rbp)
.L7:
    .loc 1 120 0 discriminator 2
    addl    $1, -5180(%rbp)
.L6:
    .loc 1 120 0 discriminator 1
    cmpl    $9, -5180(%rbp)
    jle     .L8
    .loc 1 120 0 discriminator 3
    movq    -2096(%rbp), %rax
.LBE3:
    cmpq    $1, %rax
    je      .L9
    .loc 1 120 0 discriminator 1
    movl    $0, -5188(%rbp)
    jmp     .L10
.L9:
    .loc 1 120 0 discriminator 2
    addl    $1, -5184(%rbp)
.L5:
    .loc 1 120 0 discriminator 1
    cmpl    $9, -5184(%rbp)
    jle     .L11
.L10:
    movl    -5188(%rbp), %eax
.LBE2:
    testl   %eax, %eax
    jne     .L12

```

```

        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L12:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 120 0
        movl    $0, %ecx
.LBE526:
        .loc 1 119 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 120 0 discriminator 2
        movslq  %ecx, %rdi
        movslq  %eax, %rsi
        movl    a(,%rsi,4), %ebx
        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:
        addl    $1, %eax
.LVL10:
        cmpl    $9, %eax
        jle     .L5
.LBE527:
        .loc 1 120 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 120 0 discriminator 2
        addl    $1, %ecx
.LVL11:
        cmpl    $9, %ecx

```



```

        jle      .L1295
.LVL12:
.L1038:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB528:
.LBB529:
        .loc 1 120 0
        movl    a(%rip), %r13d
        movl    a+4(%rip), %r12d
        movl    $a, %ecx
        movl    a+8(%rip), %ebp
        movl    a+12(%rip), %ebx
        movl    a+16(%rip), %r11d
        movl    a+20(%rip), %r10d
        movl    a+24(%rip), %r9d
        movl    a+28(%rip), %r8d
        movl    a+32(%rip), %edi
        movl    a+36(%rip), %esi
.LVL6:
.L5:
        movl    (%rcx), %edx
        xorl    %eax, %eax
        cmpl    %r13d, %edx
        sete    %al
.LVL7:
        xorl    %r14d, %r14d
        cmpl    %r12d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL8:
        xorl    %r14d, %r14d
        cmpl    %ebp, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL9:
        xorl    %r14d, %r14d
        cmpl    %ebx, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL10:
        xorl    %r14d, %r14d
        cmpl    %r11d, %edx
        sete    %r14b

```

```

        addq    %r14, %rax
.LVL11:
        xorl    %r14d, %r14d
        cmpl    %r10d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL12:
        xorl    %r14d, %r14d
        cmpl    %r9d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL13:
        xorl    %r14d, %r14d
        cmpl    %r8d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL14:
        xorl    %r14d, %r14d
        cmpl    %edi, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL15:
        cmpl    %esi, %edx
        sete    %dl
        movzbl  %dl, %edx
        addq    %rdx, %rax
.LVL16:
.LBE529:
        cmpq    $1, %rax
        jne     .L4164
        addq    $4, %rcx
        .loc 1 120 0 is_stmt 0 discriminator 2
        cmpq    $a+40, %rcx
        jne     .L5
.LVL17:
.L6:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %edi
        movl    $0, %eax
        call    printf
        .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    movl    $0, %eax
    call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
    .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    xorl    %eax, %eax
    call    __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O0` produces:

```

    .loc 1 120 0
#NO_APP
    movq    stdout(%rip), %rax
    movq    %rax, %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
    .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 97 0

```

```

        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al

```

```

        jne      .L1045
.L5:
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movq     -32(%rbp), %rax
        movq     %rax, %rcx
        movl     $11, %edx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq     %rbx, %rcx
        movl     $11, %edx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movl     $11, %edx
        movq     %rbx, %rcx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movq    -40(%rbp), %rax
        movl    $.LC4, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    L_buffer_printf
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        movl    $0, %eax
        call    L_buffer_printf
.LVL6:
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        xorl    %eax, %eax
        call    L_buffer_printf
.LVL279:
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    syslog
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:

```



```

        .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        xorl    %eax, %eax
        call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- assert(i >= 2); with gcc -g -O produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $1, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- TRAD_assert(i >= 2); with gcc -g -O produces:

```

        .loc 1 120 0
#NO_APP

```

```

        movl    24(%rsp), %eax
        cmpl    $1, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1
        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `I(i >= 2);` with `gcc -g -O` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $1, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `DI(i >= 2);` with `gcc -g -O` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(int i=0, i!=10, i++, a[i]>=0));` with `gcc -g -O` produces:

```

.LVL5:
#NO_APP
.LBB269:
    .loc 1 120 0
    cmpl    $0, a(%rip)
    js      .L5
    movl    $a+4, %eax
    movl    $a+40, %edx
.LVL6:
.L7:
    .loc 1 120 0 is_stmt 0 discriminator 2
    cmpl    $0, (%rax)
    js      .L5
    addq    $4, %rax
    cmpq    %rdx, %rax
    jne     .L7
.LVL7:
.L6:
.LBE269:
    .loc 1 121 0 is_stmt 1

```

- `d = now();` with `gcc -g -O` produces:

```

    .loc 1 120 0
#NO_APP
    call    now
.LVL5:
    movsd   %xmm0, 24(%rsp)
    .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    movl    $0, %eax
    call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
    .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O` produces:

```

.LVL5:

```

```

#NO_APP
.LBB526:
.LBB527:
    .loc 2 97 0
    movq    stdout(%rip), %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL6:
.LBE527:
.LBE526:
    .loc 1 121 0

```

- `DL("helloworldn");` with `gcc -g -O` produces:

```

    .loc 1 120 0
#NO_APP
    movl    $0, _dl_target(%rip)
    .loc 1 121 0

```

- `assert(i >= 2);` with `gcc -g -O` produces:

```

    .loc 1 120 0
#NO_APP
    movl    24(%rsp), %eax
    cmpl    $1, %eax
    jg      .L5
    .loc 1 120 0 is_stmt 0 discriminator 1
    movl    $120, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.LVL5:
.L5:
    .loc 1 121 0 is_stmt 1

```

- `TRAD_assert(i >= 2);` with `gcc -g -O` produces:

```

    .loc 1 120 0
#NO_APP
    movl    24(%rsp), %eax
    cmpl    $1, %eax
    jg      .L5
.LVL5:
.LBB526:
.LBB527:
    .loc 2 97 0 discriminator 1
    movl    $.LC4, %r9d

```

```

        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1
        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `I(i >= 2);` with `gcc -g -O` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $1, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `DI(i >= 2);` with `gcc -g -O` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(int i=0, i!=10, i++, a[i]>=0));` with `gcc -g -O` produces:

```

.LVL5:
#NO_APP
.LBB269:
        .loc 1 120 0
        cmpl    $0, a(%rip)
        js      .L5
        movl    $a+4, %eax

```

```

        movl    $a+40, %edx
.LVL6:
.L7:
        .loc 1 120 0 is_stmt 0 discriminator 2
        cmpl    $0, (%rax)
        js      .L5
        addq    $4, %rax
        cmpq    %rdx, %rax
        jne     .L7
.LVL7:
.L6:
.LBE269:
        .loc 1 121 0 is_stmt 1

```

- `d = now();` with `gcc -g -O` produces:

```

        .loc 1 120 0
#NO_APP
        call    now
.LVL5:
        movsd   %xmm0, 24(%rsp)
        .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 104 0
        movl    $.LC4, %esi
        movl    $1, %edi
        movl    $0, %eax
        call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi

```

```

        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `DL("helloworldn");` with `gcc -g -O` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O3` produces:

```

        .loc 1 120 0

```

- `asm("nop");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:


```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movslq 20(%rsp), %rax
        movl   a(,%rax,4), %eax
        movl   %eax, 24(%rsp)
        .loc 1 121 0

```

- `assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl   -56(%rbp), %eax
        cmpl   $9, %eax
        jg     .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl   $120, %edx
        movl   $.LC4, %esi
        movl   $.LC5, %edi
        call   _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl   24(%rsp), %eax
        cmpl   $9, %eax
        jg     .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl   $120, %edx
        movl   $.LC4, %esi
        movl   $.LC5, %edi
        call   _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl   24(%rsp), %eax
        cmpl   $9, %eax
        jle    .L274
.L5:
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- BSD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- BSD_assert(i >= 10); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- TRAD_assert(i >= 10); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- TRAD_assert(i >= 10); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1
        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O0` produces:


```

#NO_APP
.LBB2:
    .loc 1 120 0
    movl    $1, -3132(%rbp)
    movq    $str, -2096(%rbp)
    jmp     .L5
.L8:
    .loc 1 120 0 is_stmt 0 discriminator 2
    call    __ctype_b_loc
    movq    (%rax), %rdx
    movq    -2096(%rbp), %rax
    movzbl  (%rax), %eax
    movsbq  %al, %rax
    addq    %rax, %rax
    addq    %rdx, %rax
    movzwl  (%rax), %eax
    movzwl  %ax, %eax
    andl    $512, %eax
    testl   %eax, %eax
    jne     .L6
    .loc 1 120 0 discriminator 1
    movl    $0, -3132(%rbp)
    jmp     .L7
.L6:
    .loc 1 120 0 discriminator 2
    addq    $1, -2096(%rbp)
.L5:
    .loc 1 120 0 discriminator 1
    movq    -2096(%rbp), %rax
    movzbl  (%rax), %eax
    testb   %al, %al
    jne     .L8
.L7:
    movl    -3132(%rbp), %eax
.LBE2:
    testl   %eax, %eax
    jne     .L9
    movl    $120, %edx
    movl    $.LC4, %esi
    movl    $.LC5, %edi
    call    _I_default_handler
.L9:
    .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB269:

```

```

        .loc 1 120 0
        movzbl  str(%rip), %ebx
        testb  %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
.LVL7:
.L7:
        .loc 1 120 0 is_stmt 0 discriminator 2
        movsbq  %bl, %rbx
        testb   $2, 1(%rdx,%rbx,2)
        je      .L6
        addq    $1, %rax
.LVL8:
        movzbl  (%rax), %ebx
        testb   %bl, %bl
        jne     .L7
.LVL9:
.L5:
.LBE269:
        .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O3` produces:

```

.LVL5:
#NO_APP
.LBB271:
        .loc 1 120 0
        movsbq  str(%rip), %rbx
        testb   %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
        jmp     .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 120 0 is_stmt 0 discriminator 2
        addq    $1, %rax
.LVL8:
        movsbq  (%rax), %rbx
        testb   %bl, %bl
        je      .L5
.LVL9:

```

```

.L7:
    testb    $2, 1(%rdx,%rbx,2)
    jne      .L2074
.LVL10:
.LBE271:
    .loc 1 120 0
    movl     $120, %edx
    movl     $.LC10, %esi
    movl     $.LC11, %edi
    call     _I_default_handler
.LVL11:
.L5:
    .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
    .loc 1 120 0
    movl     $1, -5188(%rbp)
    movl     $0, -5184(%rbp)
    jmp      .L5
.L11:
.LBB3:
    .loc 1 120 0 is_stmt 0 discriminator 2
    movq     $0, -2096(%rbp)
    movl     $0, -5180(%rbp)
    jmp      .L6
.L8:
    movl     -5184(%rbp), %eax
    cltq
    movl     a(,%rax,4), %edx
    movl     -5180(%rbp), %eax
    cltq
    movl     a(,%rax,4), %eax
    cmpl     %eax, %edx
    jne      .L7
    .loc 1 120 0 discriminator 1
    addq     $1, -2096(%rbp)
.L7:
    .loc 1 120 0 discriminator 2
    addl     $1, -5180(%rbp)
.L6:
    .loc 1 120 0 discriminator 1
    cmpl     $9, -5180(%rbp)
    jle      .L8
    .loc 1 120 0 discriminator 3
    movq     -2096(%rbp), %rax
.LBE3:

```

```

        cmpq    $1, %rax
        je      .L9
        .loc 1 120 0 discriminator 1
        movl    $0, -5188(%rbp)
        jmp     .L10
.L9:
        .loc 1 120 0 discriminator 2
        addl    $1, -5184(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        cmpl    $9, -5184(%rbp)
        jle     .L11
.L10:
        movl    -5188(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L12
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L12:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 120 0
        movl    $0, %ecx
.LBE526:
        .loc 1 119 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 120 0 discriminator 2
        movslq   %ecx, %rdi
        movslq   %eax, %rsi
        movl     a(,%rsi,4), %ebx

```

```

        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:
        addl    $1, %eax
.LVL10:
        cmpl    $9, %eax
        jle     .L5
.LBE527:
        .loc 1 120 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 120 0 discriminator 2
        addl    $1, %ecx
.LVL11:
        cmpl    $9, %ecx
        jle     .L1295
.LVL12:
.L1038:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB528:
.LBB529:
        .loc 1 120 0
        movl    a(%rip), %r13d
        movl    a+4(%rip), %r12d
        movl    $a, %ecx
        movl    a+8(%rip), %ebp
        movl    a+12(%rip), %ebx
        movl    a+16(%rip), %r11d
        movl    a+20(%rip), %r10d
        movl    a+24(%rip), %r9d
        movl    a+28(%rip), %r8d
        movl    a+32(%rip), %edi
        movl    a+36(%rip), %esi
.LVL6:
.L5:
        movl    (%rcx), %edx
        xorl    %eax, %eax
        cmpl    %r13d, %edx
        sete    %al
.LVL7:
        xorl    %r14d, %r14d

```

```

        cmpl    %r12d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL8:
        xorl    %r14d, %r14d
        cmpl    %ebp, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL9:
        xorl    %r14d, %r14d
        cmpl    %ebx, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL10:
        xorl    %r14d, %r14d
        cmpl    %r11d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL11:
        xorl    %r14d, %r14d
        cmpl    %r10d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL12:
        xorl    %r14d, %r14d
        cmpl    %r9d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL13:
        xorl    %r14d, %r14d
        cmpl    %r8d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL14:
        xorl    %r14d, %r14d
        cmpl    %edi, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL15:
        cmpl    %esi, %edx
        sete    %dl
        movzbl  %dl, %edx
        addq    %rdx, %rax
.LVL16:
.LBE529:
        cmpq    $1, %rax
        jne     .L4164
        addq    $4, %rcx
        .loc 1 120 0 is_stmt 0 discriminator 2
        cmpq    $a+40, %rcx

```

```

        jne      .L5
.LVL17:
.L6:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl     $.LC4, %edi
        movl     $0, %eax
        call     printf
        .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 104 0
        movl     $.LC4, %esi
        movl     $1, %edi
        movl     $0, %eax
        call     __printf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 104 0
        movl     $.LC4, %esi
        movl     $1, %edi
        xorl     %eax, %eax
        call     __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- L("helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- DL("helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```


- `DL("helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `DL("helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi

```

```

        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1

```

```

        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `LHP(fprintf,log,"helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movq    -32(%rbp), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
        .loc 1 121 0

```

- `LHP(fprintf,log,"helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq    %rbx, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```
.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 97 0
    movl    $11, %edx
    movq    %rbx, %rcx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```
    .loc 1 120 0
#NO_APP
    movq    -40(%rbp), %rax
    movl    $.LC4, %esi
    movq    %rax, %rdi
    movl    $0, %eax
    call    L_buffer_printf
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```
    .loc 1 120 0
#NO_APP
    movl    $.LC4, %esi
    movq    %rbx, %rdi
    movl    $0, %eax
    call    L_buffer_printf
.LVL6:
    .loc 1 121 0
```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```
    .loc 1 120 0
#NO_APP
    movl    $.LC4, %esi
    movq    %rbx, %rdi
    xorl    %eax, %eax
    call    L_buffer_printf
.LVL279:
    .loc 1 121 0
```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    syslog
        .loc 1 121 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        xorl    %eax, %eax
        call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- `asm("");` with `gcc -g -O0` produces:

```

        .loc 1 120 0

```

- `asm("");` with `gcc -g -O1` produces:

```
.loc 1 120 0
```

- `asm("");` with `gcc -g -O3` produces:

```
.loc 1 120 0
```

- `asm("nop");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```


- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movslq  20(%rsp), %rax
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `assert(i >= 10)`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- `BSD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- `TRAD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1

```

```

        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `assert(i >= 10);` with `gcc -g -O` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `BSD_assert(i >= 10);` with `gcc -g -O` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1
        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `DI(i >= 10);` with `gcc -g -O` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```


- `asm("");` with `gcc -g -O0` produces:

```
.loc 1 120 0
```

- `asm("");` with `gcc -g -O1` produces:

```
.loc 1 120 0
```

- `asm("");` with `gcc -g -O3` produces:

```
.loc 1 120 0
```

- `asm("nop");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop");` with `gcc -g -O3` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O0` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O1` produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
    nop;nop;
# 0 "" 2
```

- `asm("nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `asm("nop;nop;nop;nop;nop;");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
# 120 "tmp.c" 1
        nop;nop;nop;nop;nop;
# 0 "" 2

```

- `i = 4;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, -56(%rbp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `i = 4;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $4, 24(%rsp)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `gi = 11`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $11, gi(%rip)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, -48(%rbp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, 28(%rsp)
        .loc 1 121 0

```

- `f = 12.0`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, 28(%rsp)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    .LC4(%rip), %eax
        movl    %eax, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movss   .LC4(%rip), %xmm0
        movss   %xmm0, gf(%rip)
        .loc 1 121 0

```

- `gf = 12.0;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0x41400000, gf(%rip)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        addl    $1, %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `i++;` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        addl    $1, %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `gi++;` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++;` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `gi++`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    gi(%rip), %eax
        addl    $1, %eax
        movl    %eax, gi(%rip)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -60(%rbp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, -56(%rbp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    20(%rsp), %eax
        cltq
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `j = a[i]`; with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movslq  20(%rsp), %rax
        movl    a(,%rax,4), %eax
        movl    %eax, 24(%rsp)
        .loc 1 121 0

```

- `assert(i >= 10)`; with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $.LC4, %edx
        movl    $120, %esi
        movl    $.LC5, %edi
        call    __BSD_assert
.L5:
        .loc 1 121 0 is_stmt 1

```

- `BSD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `BSD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stderr(%rip), %rax
        movl    $.LC4, %r8d
        movl    $120, %ecx
        movl    $.LC5, %edx
        movl    $.LC6, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    fprintf
        movl    $1, %edi
        call    exit
.L5:
        .loc 1 121 0 is_stmt 1

```

- `TRAD_assert(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movl    $.LC4, %r9d
        movl    $120, %r8d
        movl    $.LC5, %ecx
        movl    $.LC6, %edx
        movl    $1, %esi
        movq    stderr(%rip), %rdi
        movl    $0, %eax
        call    __fprintf_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 120 0 discriminator 1

```



```

        movl    $1, %edi
        call    exit
.LVL7:
.L5:
        .loc 1 121 0

```

- `TRAD_assert(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L21
        .loc 1 121 0

```

- `I(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    -56(%rbp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jg      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.LVL5:
.L5:
        .loc 1 121 0 is_stmt 1

```

- `I(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    24(%rsp), %eax
        cmpl    $9, %eax
        jle     .L274
.L5:
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `DI(i >= 10);` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _di_target(%rip)
        .loc 1 121 0

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O0` produces:

```

#NO_APP
.LBB2:
        .loc 1 120 0
        movl    $1, -3132(%rbp)
        movq    $str, -2096(%rbp)
        jmp     .L5
.L8:
        .loc 1 120 0 is_stmt 0 discriminator 2
        call    __ctype_b_loc
        movq    (%rax), %rdx
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        movsbq  %al, %rax
        addq    %rax, %rax
        addq    %rdx, %rax
        movzwl  (%rax), %eax
        movzwl  %ax, %eax

```

```

        andl    $512, %eax
        testl   %eax, %eax
        jne     .L6
        .loc 1 120 0 discriminator 1
        movl    $0, -3132(%rbp)
        jmp     .L7
.L6:
        .loc 1 120 0 discriminator 2
        addq    $1, -2096(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
        movq    -2096(%rbp), %rax
        movzbl  (%rax), %eax
        testb   %al, %al
        jne     .L8
.L7:
        movl    -3132(%rbp), %eax
.LBE2:
        testl   %eax, %eax
        jne     .L9
        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L9:
        .loc 1 121 0 is_stmt 1

```

- `I(A(char *p = str, *p != '\0', p++, islower(*p)))`; with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB269:
        .loc 1 120 0
        movzbl  str(%rip), %ebx
        testb   %bl, %bl
        je      .L5
        call    __ctype_b_loc
.LVL6:
        movq    (%rax), %rdx
        movl    $str, %eax
.LVL7:
.L7:
        .loc 1 120 0 is_stmt 0 discriminator 2
        movsbq   %bl, %rbx
        testb    $2, 1(%rdx,%rbx,2)
        je       .L6
        addq     $1, %rax
.LVL8:
        movzbl  (%rax), %ebx

```

```

        testb    %b1, %b1
        jne      .L7
.LVL9:
.L5:
.LBE269:
        .loc 1 121 0 is_stmt 1

```

- I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB271:
        .loc 1 120 0
        movsbq   str(%rip), %rbx
        testb    %b1, %b1
        je       .L5
        call     __ctype_b_loc
.LVL6:
        movq     (%rax), %rdx
        movl     $str, %eax
        jmp      .L7
.LVL7:
        .p2align 4,,10
        .p2align 3
.L2074:
        .loc 1 120 0 is_stmt 0 discriminator 2
        addq     $1, %rax
.LVL8:
        movsbq   (%rax), %rbx
        testb    %b1, %b1
        je       .L5
.LVL9:
.L7:
        testb    $2, 1(%rdx,%rbx,2)
        jne      .L2074
.LVL10:
.LBE271:
        .loc 1 120 0
        movl     $120, %edx
        movl     $.LC10, %esi
        movl     $.LC11, %edi
        call     _I_default_handler
.LVL11:
.L5:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))); with gcc -g -O0 produces:

```

#NO_APP
.LBB2:
    .loc 1 120 0
    movl    $1, -5188(%rbp)
    movl    $0, -5184(%rbp)
    jmp     .L5
.L11:
.LBB3:
    .loc 1 120 0 is_stmt 0 discriminator 2
    movq    $0, -2096(%rbp)
    movl    $0, -5180(%rbp)
    jmp     .L6
.L8:
    movl    -5184(%rbp), %eax
    cltq
    movl    a(,%rax,4), %edx
    movl    -5180(%rbp), %eax
    cltq
    movl    a(,%rax,4), %eax
    cmpl    %eax, %edx
    jne     .L7
    .loc 1 120 0 discriminator 1
    addq    $1, -2096(%rbp)
.L7:
    .loc 1 120 0 discriminator 2
    addl    $1, -5180(%rbp)
.L6:
    .loc 1 120 0 discriminator 1
    cmpl    $9, -5180(%rbp)
    jle     .L8
    .loc 1 120 0 discriminator 3
    movq    -2096(%rbp), %rax
.LBE3:
    cmpq    $1, %rax
    je      .L9
    .loc 1 120 0 discriminator 1
    movl    $0, -5188(%rbp)
    jmp     .L10
.L9:
    .loc 1 120 0 discriminator 2
    addl    $1, -5184(%rbp)
.L5:
    .loc 1 120 0 discriminator 1
    cmpl    $9, -5184(%rbp)
    jle     .L11
.L10:
    movl    -5188(%rbp), %eax
.LBE2:
    testl   %eax, %eax
    jne     .L12

```

```

        movl    $120, %edx
        movl    $.LC4, %esi
        movl    $.LC5, %edi
        call    _I_default_handler
.L12:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
        .loc 1 120 0
        movl    $0, %ecx
.LBE526:
        .loc 1 119 0
        movl    $0, %eax
        movl    $0, %edx
        jmp     .L5
.LVL6:
.L1295:
.LBB528:
        movl    $0, %eax
.LVL7:
        movl    $0, %edx
.LVL8:
.L5:
.LBB527:
        .loc 1 120 0 discriminator 2
        movslq  %ecx, %rdi
        movslq  %eax, %rsi
        movl    a(,%rsi,4), %ebx
        cmpl    %ebx, a(,%rdi,4)
        sete    %sil
        movzbl  %sil, %esi
        addq    %rsi, %rdx
.LVL9:
        addl    $1, %eax
.LVL10:
        cmpl    $9, %eax
        jle     .L5
.LBE527:
        .loc 1 120 0 is_stmt 0 discriminator 3
        cmpq    $1, %rdx
        jne     .L8
        .loc 1 120 0 discriminator 2
        addl    $1, %ecx
.LVL11:
        cmpl    $9, %ecx

```

```

        jle      .L1295
.LVL12:
.L1038:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j]))));
with gcc -g -O3 produces:

```

.LVL5:
#NO_APP
.LBB528:
.LBB529:
        .loc 1 120 0
        movl    a(%rip), %r13d
        movl    a+4(%rip), %r12d
        movl    $a, %ecx
        movl    a+8(%rip), %ebp
        movl    a+12(%rip), %ebx
        movl    a+16(%rip), %r11d
        movl    a+20(%rip), %r10d
        movl    a+24(%rip), %r9d
        movl    a+28(%rip), %r8d
        movl    a+32(%rip), %edi
        movl    a+36(%rip), %esi
.LVL6:
.L5:
        movl    (%rcx), %edx
        xorl    %eax, %eax
        cmpl    %r13d, %edx
        sete    %al
.LVL7:
        xorl    %r14d, %r14d
        cmpl    %r12d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL8:
        xorl    %r14d, %r14d
        cmpl    %ebp, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL9:
        xorl    %r14d, %r14d
        cmpl    %ebx, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL10:
        xorl    %r14d, %r14d
        cmpl    %r11d, %edx
        sete    %r14b

```

```

        addq    %r14, %rax
.LVL11:
        xorl    %r14d, %r14d
        cmpl    %r10d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL12:
        xorl    %r14d, %r14d
        cmpl    %r9d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL13:
        xorl    %r14d, %r14d
        cmpl    %r8d, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL14:
        xorl    %r14d, %r14d
        cmpl    %edi, %edx
        sete    %r14b
        addq    %r14, %rax
.LVL15:
        cmpl    %esi, %edx
        sete    %dl
        movzbl  %dl, %edx
        addq    %rdx, %rax
.LVL16:
.LBE529:
        cmpq    $1, %rax
        jne     .L4164
        addq    $4, %rcx
        .loc 1 120 0 is_stmt 0 discriminator 2
        cmpq    $a+40, %rcx
        jne     .L5
.LVL17:
.L6:
.LBE528:
        .loc 1 121 0 is_stmt 1

```

- `printf("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %edi
        movl    $0, %eax
        call    printf
        .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O1` produces:


```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    movl    $0, %eax
    call    __printf_chk
.LVL6:
.LBE527:
.LBE526:
    .loc 1 121 0

```

- `printf("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
    .loc 2 104 0
    movl    $.LC4, %esi
    movl    $1, %edi
    xorl    %eax, %eax
    call    __printf_chk
.LVL278:
.LBE1051:
.LBE1050:
    .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O0` produces:

```

    .loc 1 120 0
#NO_APP
    movq    stdout(%rip), %rax
    movq    %rax, %rcx
    movl    $11, %edx
    movl    $1, %esi
    movl    $.LC4, %edi
    call    fwrite
    .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O1` produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
    .loc 2 97 0

```

```

        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `L("helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- `DL("helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `DL("helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `DL("helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, _dl_target(%rip)
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = 0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $0, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        jne     .L1045
.L5:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O0` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        andl    $16, %eax
        testl   %eax, %eax
        je      .L5
        .loc 1 120 0 is_stmt 0 discriminator 1
        movq    stdout(%rip), %rax
        movq    %rax, %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.L5:
        .loc 1 121 0 is_stmt 1

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O1` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al
        je      .L5
.LVL5:
.LBB526:
.LBB527:
        .loc 2 97 0 discriminator 1
        movq    stdout(%rip), %rcx
        movl    $11, %edx
        movl    $1, %esi
        movl    $.LC4, %edi
        call    fwrite
.LVL6:
.L5:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `gi = ~0; LG(gi & 0x10, "helloworldn");` with `gcc -g -O3` produces:

```

        .loc 1 120 0
#NO_APP
        movl    $-1, gi(%rip)
        movl    gi(%rip), %eax
        testb   $16, %al

```

```

        jne      .L1045
.L5:
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movq     -32(%rbp), %rax
        movq     %rax, %rcx
        movl     $11, %edx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:
        .loc 2 97 0
        movq     %rbx, %rcx
        movl     $11, %edx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 2 97 0
        movl     $11, %edx
        movq     %rbx, %rcx
        movl     $1, %esi
        movl     $.LC4, %edi
        call     fwrite
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movq    -40(%rbp), %rax
        movl    $.LC4, %esi
        movq    %rax, %rdi
        movl    $0, %eax
        call    L_buffer_printf
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        movl    $0, %eax
        call    L_buffer_printf
.LVL6:
        .loc 1 121 0

```

- LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movq    %rbx, %rdi
        xorl    %eax, %eax
        call    L_buffer_printf
.LVL279:
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O0 produces:

```

        .loc 1 120 0
#NO_APP
        movl    $.LC4, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    syslog
        .loc 1 121 0

```

- LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

```

.LVL5:
#NO_APP
.LBB526:
.LBB527:

```

```

        .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        movl    $0, %eax
        call    __syslog_chk
.LVL6:
.LBE527:
.LBE526:
        .loc 1 121 0

```

- `LHP(syslog,LOG_USER,"helloworldn");` with `gcc -g -O3` produces:

```

.LVL277:
#NO_APP
.LBB1050:
.LBB1051:
        .loc 3 31 0
        movl    $.LC4, %edx
        movl    $1, %esi
        movl    $8, %edi
        xorl    %eax, %eax
        call    __syslog_chk
.LVL278:
.LBE1051:
.LBE1050:
        .loc 1 121 0

```

4 Conclusion

Finally, if you have used this package on an interesting (or uninteresting) architecture please mail me a copy of the results for the nana home page.